



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STROJNÍHO INŽENÝRSTVÍ

FACULTY OF MECHANICAL ENGINEERING

**ÚSTAV MECHANIKY TĚLES, MECHATRONIKY A
BIOMECHANIKY**

INSTITUTE OF SOLID MECHANICS, MECHATRONICS AND BIOMECHANICS

**VYUŽITÍ UMĚLÉ INTELIGENCE K MONITOROVÁNÍ STAVU
OBRÁBĚCÍHO STROJE**

USING ARTIFICIAL INTELLIGENCE TO MONITOR THE STATE OF THE MACHINE

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Jan Kubisz

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Jiří Kovář, Ph.D.

BRNO 2020

Zadání diplomové práce

Ústav: Ústav mechaniky těles, mechatroniky a biomechaniky
Student: **Bc. Jan Kubisz**
Studijní program: Aplikované vědy v inženýrství
Studijní obor: Mechatronika
Vedoucí práce: **Ing. Jiří Kovář, Ph.D.**
Akademický rok: 2019/20

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

Využití umělé inteligence k monitorování stavu obráběcího stroje

Stručná charakteristika problematiky úkolu:

Cílem práce je použití existující platformy obsahující některé metody tzv. umělé inteligence k monitorování stavu stroje vyhodnocováním měřených dat.

Cíle diplomové práce:

Osvojte si práci s neuronovými sítěmi a platformou UVSSR Neural Network
Navrhněte neuronovou síť pro účely monitorování technického stavu stroje
Ověřte navrženou síť v reálných podmínkách

Seznam doporučené literatury:

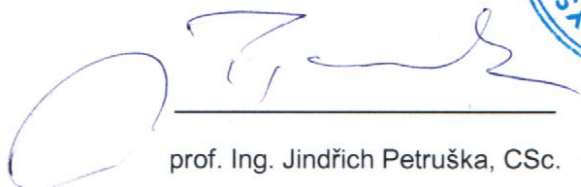
MITCHELL, T. M., Machine Learning, McGraw-Hill Science/Engineering/Math, 1997, ISBN: 0070428077

BISHOP Ch. M., Pattern Recognition and Machine Learning, Springer S+M Media, 2006, ISBN: 0-38-31073-8

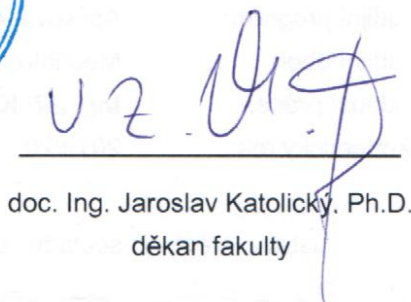
Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2019/20

V Brně, dne 23. 10. 2019





prof. Ing. Jindřich Petruška, CSc.
ředitel ústavu



doc. Ing. Jaroslav Katolický, Ph.D.
děkan fakulty

Abstrakt

Diplomová práce se zabývá vytvořením vnitřní struktury neuronových sítí s cílem vytvoření umělé neuronové sítě schopné monitorovat stav stroje a vyhodnocovat jeho zbývající život. Cílem je vytvoření algoritmů a knihovny pro návrh a učení neuronové sítě, a tím i hlubší pochopení problematiky než při použití existujících knihoven. Zvolenou metodou byla dopředná síť s architekturou vícevrstvého perceptronu a učenou pomocí zpětné propagace. Dosaženým výsledkem bylo, že síť byla schopna určovat stav součásti z měření vibrací a na jeho základě byl odhadován zbývající život dané součásti.

Abstract

Diploma thesis focus on creation of neural network's internal structure with goal of creation Artificial Neural Network capable of machine state monitoring and predicting its remaining usefull life. Main goal is creation of algorithm's and library for design and learning of Artificial Neural Network, and deeper understanding of the problematics in the process, then by utilising existing libraries. Selected method was forward-propagation network with multi-layered perceptron architecture, and backpropagation learning. Achieved results was, that the network was able to determine parts state from vibration measurement and on its basis predict remaining usefull life.

Klíčová slova

Neuronová síť, Učení s učitelem, Návrh sítě, Chybové funkce, Odhad zbývajícího života, Monitorování chodu stroje, Dopředná propagace, Zpětná propagace, Prořezávání sítě, predikce chyb

Key Words

Neural Network, Supervised learning, Network design, Error functions, Remaining usefull life, Machine operation monitoring, Forward propagation, Backpropagation, Pruning, Failure prediction

Čestné prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval sám na základě svých znalostí, snahy, dovedností, rad a pokynů vedoucího diplomové práce. Uvedl jsem veškeré použité podklady a literaturu.

V Brně dne:

.....

Bc. Jan Kubisz

Poděkování

Tímto bych rád poděkoval vedoucímu mé diplomové práce Ing. Jiřímu Kováři, Ph.D. za rady a pozitivní přístup v průběhu zpracovávání práce. Dále bych chtěl poděkovat svým rodičům za podporu a pochopení v průběhu studia.

Jan Kubisz

Bibliografická citace

KUBISZ, Jan. *Využití umělé inteligence k monitorování stavu obráběcího stroje*. Brno, 2020. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/124628>. Diplomová práce. Vysoké učení technické v Brně, Fakulta strojního inženýrství, Ústav mechaniky těles, mechatroniky a biomechaniky. Vedoucí práce Ing. Jiří Kovář, Ph.D.

Obsah

1. Úvod	11
2. Formulace problému a cíle řešení	13
3. Představení tématu	15
3.1. Monitorování stavu strojů	15
3.2. Aplikace UI	16
4. Umělé neuronové sítě	17
4.1. Struktura neuronové sítě	17
4.2. Učení sítě	18
4.2.1. Učení s učitelem	18
4.3. Aktivační funkce	19
4.4. Chybové funkce	20
4.4.1. MSE	20
4.4.2. MAE	20
4.4.3. Pseudo-Huber Loss	20
4.4.4. Quantile Loss	21
4.5. Backpropagation	22
4.6. Learning rate	24
4.7. Příprava dat	25
4.8. Vyhodnocování dat	26
4.9. Volba sítě	28
4.9.1. Prořezávání sítě	29
4.10. Odhad zbývajících života	31
4.10.1. Run-to-Failure metoda	32
4.10.2. Threshold metoda	33
4.10.3. Klasifikátor skupin	33
5. Vypracování tvorby sítě	35
5.1. Volba a zpracování dat	37
5.1.1. Data pro první cyklus učení	41
5.1.2. Data pro druhý cyklus učení	41
5.2. Vytvoření neuronové sítě	43
5.2.1. Definování sítě	43
5.2.2. Učení sítě	44
5.3. RUL	47
5.3.1. Run-to-Failure Data	47
5.3.2. Threshold data	48
5.3.3. RUL dle skupin	50
5.3.4. Vyhodnocení RUL	52
5.3.5. Doučování sítě	53

6. Zhodnocení výsledků a diskuze	55
7. Závěr	57
8. Seznam zdrojů	59
8.1. Použitá literatura	59
8.2. Obrázky	62
9. Seznam zkratk, symbolů a obrázků	63
9.1. Seznam symbolů	63
9.2. Seznam zkratk	64
9.3. Seznam obrázků	65

1. Úvod

Údržba strojů je nedílnou součástí pracovního cyklu a je nezbytná pro udržení jejich provozuschopnosti. Jedním z hlavních úkolů údržby je zachování, nebo i zvyšování, spolehlivosti stroje jak z pohledu provozních vlastností, tak z pohledu omezení prostojů v důsledku jeho poruch. Předcházení těmto poruchám tvoří součást tzv. proaktivní, nebo prediktivní údržby. Efektivní prediktivní údržba poté dovoluje optimalizovat, a tím úměrně snižovat, s ní spojené finanční náklady. Tato skutečnost je jedním z hlavních motivátorů pro vytváření a zavádění nástrojů a postupů, které by umožňovaly reagovat na poruchy dopředu, nikoli následně.

Prvním krokem k zavedení těchto postupů a nástrojů je implementace diagnostických zařízení využívajících různých měřicích metod a technik pro určení co nejpřesnějšího aktuálního technického stavu stroje. Toto vyhodnocování se děje na základě měřených dat z daného stroje. Ve strojírenství je jednou z nejběžnějších metod snímání vibrací kritických součástí stroje a vyhodnocování jejich stavu buďto personálem, nebo automatizovaným systémem. S rozvojem mechatroniky jsou, typově stejné stroje, stále více senzoricky vybaveny, což má přímý důsledek v nárůstu objemu zaznamenaných dat. Vyhodnocení a kontextualizace dat je ale stále více problematická, hlavně z důvodu složitosti strojů. To je jedním z důvodů, proč se v této oblasti prosazují metody umělé inteligence, přičemž stále populárnějším systémem jsou umělé neuronové sítě. Při jejich vhodné aplikaci je možno optimalizovat plánování preventivní údržby s ohledem na maximální životnost součástí a zároveň předcházet neočekávaným havarijním stavům.

Tato diplomová práce se zabývá možným využitím umělých neuronových sítí pro vyhodnocování vibrací součástí stroje, určení stavu dané součásti a vytvoření odhadu zbývajících života pro prediktivní údržbu. Princip použití umělé neuronové sítě je založen na napodobování spojů lidského mozku. Jednotlivé prvky této umělé sítě se nazývají neurony, a obdobně jako v mozku, jsou vzájemně propojeny ve vrstvách a vytváří tak různé struktury funkcí specifické ku požadovanému procesu. Použití neuronových sítí umožňuje vytvoření popisu systému pouze na základě vstupních a výstupních dat. Jedná se tedy o „black box“ model schopný popsat i složité fyzikální děje bez znalosti jejich přesných spojitostí. To přináší výhody v možnosti jednoduchého napodobení i velmi složitých systémů, ovšem i nevýhody v neznalosti dění uvnitř struktury.

V teoretické části je rozebrána struktura neuronové sítě. Jsou popsány metody, které budou použity a jejich jednotlivé části, které byly následně implementovány do vytvořené knihovny. Je nastíněna struktura vytvořeného algoritmu učení sítě a jsou vybrány metody odhadu zbývajících života.

V praktické části je popsáno vytvoření a učení neuronové sítě pro vyhodnocení měřených vibrací součástí. V práci byla vytvořena a testována architektura sítě MPL (*Multi Layered Preceptron*) s učením pomocí zpětné propagace chyby (*Backpropagation*), jejíž výstupem byl odhad stavu součástí, konkrétně ložiska. Na základě tohoto odhadu, vytvořeného pomocí neuronové sítě, byl vytvořen odhad

zbývajících života součástí. V původním záměru bylo ověření tohoto software na jednom ze strojů na pracovišti UVSSR, to bohužel nebylo možno, vzhledem k nastalé zdravotní krizi, tudíž byla použita sada veřejně přístupných dat.

V poslední části práce jsou diskutovány dosažené výsledky, jsou zhodnoceny a jsou prezentovány náměty pro další práci.

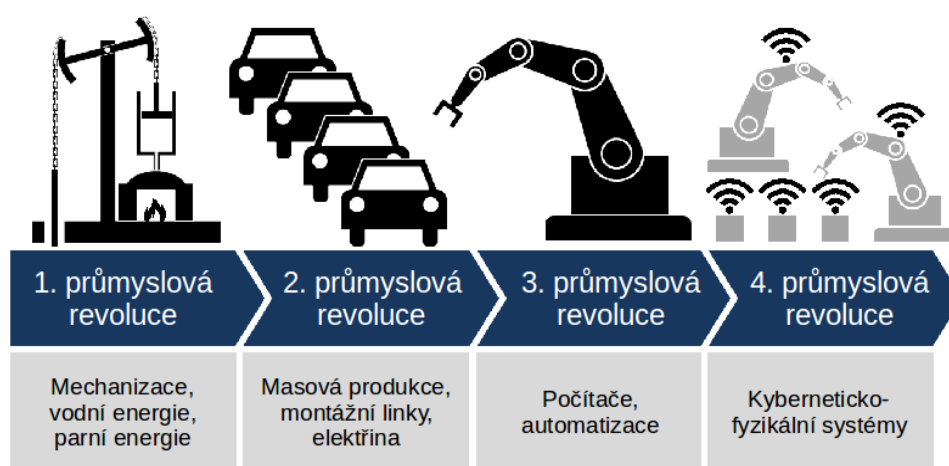
2. Formulace problému a cíle řešení

Problematika zajištění včasné údržby a chodu stroje vyžaduje odhad aktuálního stavu stroje a předpovídání výskytu možné poruchy. Toho je možno dosáhnout například měřením vibrací na dané součásti a analýzou tohoto záznamu. Provádět analýzu vibračního záznamu ručně je ovšem zdlouhavé, čímž četnost měření a tím i pravděpodobnost včasného odhalení poruchy, výrazně klesá. Proto jsou využívány umělé neuronové sítě k vyhodnocování těchto záznamů.

Cílem práce bylo vypracování knihovny a jejích součástí pro vytvoření a učení neuronové sítě, a seznámení se s problematikou její tvorby, nastavení a ladění. Následná aplikace navržené sítě na reálný datový záznam z měření vibrací, jejich zpracování a naučení sítě vyhodnotit stav součásti z tohoto záznamu. Provést otestování různých topologií sítě s cílem nalézt nejvíce vyhovující a následně ověřit funkčnost vytvořené umělé sítě na testovacím souboru dat a vyhodnocení zbývajících života stroje.

3. Představení tématu

S příchodem Průmyslu 4.0 a se stále se zvyšující dostupností levných výpočetních jednotek a senzorů, roste snaha o monitorování stále více procesů. Data z těchto procesů jsou ukládána a mohou být sdílána přes cloudová úložiště pro online sledování pohybu materiálu, výrobků, nebo stavu jednotlivých strojů, optimalizaci skladování materiálu, nebo procesů při jeho zpracování. Snaha je rovněž i o automatizaci jednodušších činností, které byly dosud vykonávány lidmi.



Obr. 1. Stupně průmyslových revolucí

3.1. Monitorování stavu strojů

Monitorování aktuálního stavu strojů je jedním z důležitých prvků ve výrobě, z důvodu snahy o maximalizaci efektivity, a tedy i minimalizaci prostoje stroje. Při současném trendu optimalizace a zeštíhlování je tedy snaha o sběr dat z kritických míst stroje, jejich analýza a predikce potenciálních poruch a jejich předcházení.

Zpracování a kontextualizace těchto dat je ovšem náročná, zvláště pokud se jedná o komplexní systémy, a tak jsou většinou stále využívány strategie preventivní, nebo reaktivní údržby. To ovšem přináší nevýhody v podobě větších nákladů na údržbu a potenciální výměnu zdravých dílů, nebo naopak dlouhé prostoje při neočekávaných poruchách. V majoritní většině procesů, které využívají akvizice dat pro prediktivní údržbu, jsou data stále vyhodnocována technikem s testovacími nástroji a pouze asi 1/3 využívá bezdrátového monitorování. Mezi hlavní tři preferované praktiky monitorování strojů se poté řadí [1][2]:

- *Analýza mazání stroje – např.: stav oleje*
- *Analýza tepelné zátěže – např.: Infračervené tepelné zobrazování*
- *Analýza vibrací – např.: snímání vibrací ložisek*

Touto snahou o monitorování procesů a stavů strojů vznikla potřeba vhodné kontextualizace získaných dat. Většinu dat ovšem nelze jednoduše interpretovat přímo ze získaných výsledků a je potřeba komplexnějšího zpracování. K tomu bývají využívány, zvláště pro komplexnější data, specializované algoritmy, nebo modely. Jednodušší aplikace je možno obvykle reprezentovat linearizovaným dynamickým modelem, který napodobuje daný systém s přesností dostatečnou pro jeho řízení, nebo vyhodnocení.

Nevýhodou spočívající v tomto způsobu napodobení systému je nutná linearizace modelu. Pokud se tedy v popisovaném systému vyskytuje příliš velká nelinearita, může být použití takového modelu nevhodné. Rovněž pokud je popisovaný děj, z fyzikálního hlediska, příliš komplexní, nebo se v popisovaném systému vyskytuje příliš mnoho proměnných a tvorba modelu kompletním rozbořením daného systému by byla příliš náročná a neefektivní, je nutné zvolit jinou variantu řešení.

Zde se, pro komplexnější data s větší náročností na zpracování, nabízí možnost použití metod umělé inteligence pro vhodnou interpretaci dat.

3.2. Aplikace UI

Umělé neuronové sítě (*Artificial Neural Network*) mohou pomoci výše zmíněný problém interpretace dat řešit. Tvoří možnou náhradu metod, které vyžadují přesný fyzikální popis děje. Výhodou použití umělých neuronových sítí je, že nevyžadují přesnou znalost jejich fyzikální podstaty popisovaných procesů. Neuronová síť se chová jako „black box“ model, kdy sama, na základě vstupních a výstupních dat, určí vnitřní parametry popisovaného systému.

Výhodou neuronových sítí je tedy to, že pro jejich aplikaci není potřeba přesná znalost fyzikálních procesů za monitorovaným procesem, ale pouze dostatečné množství dat popisujících daný proces a jeho vstupy a výstupy. Další z výhod neuronových sítí je možnost doučování sítě, a udržování tak jejího aktuálního stavu vůči možným změnám na datech.

Nevýhodou neuronových sítí je to, že u řešení není obvykle zaručena stabilita modelu a nemohou tedy být použity v kritických aplikacích.

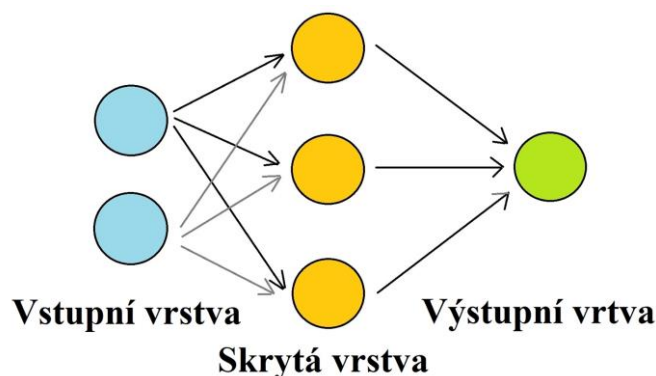
Struktura a princip neuronových sítí, s primárním zájmem na architekturu sítě MPL (*Multi Layered Perceptron*), jejich aplikace a základní architektury budou popsány v následující kapitole. V praktické části bude popsána zvolené aplikace atd...

4. Umělé neuronové sítě

Pro tuto práci byla zvolena metoda dopředné sítě (*Feed-Forward*) s architekturou MLP, jejíž prvky budou na následujících stránkách rozebrány a popsány, společně s analýzou výsledků a vyhodnocením zbývajících života testované součásti. Vybraná architektura je jednou z nejčastěji používaných variant dopředné sítě, její použití rovněž poskytlo uspokojivější výsledky než ostatní testované varianty.

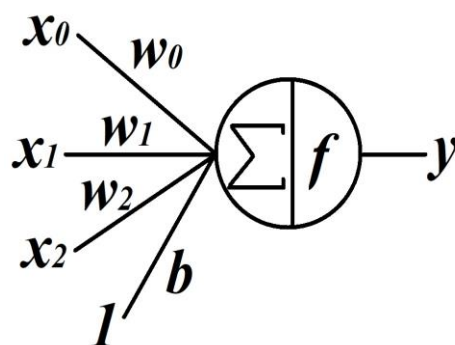
4.1. Struktura neuronové sítě

Umělá neuronová síť je rozdělena do 3 základních částí, vstupní, skryté a výstupní vrstvy, (viz. Obr. 2.) Podle zvolené struktury sítě, jsou vrstvy mezi sebou vzájemně plně, nebo částečně, propojeny vahami (šipky v Obr. 2.). Tyto propojení přenáší hodnoty mezi neurony a velikosti těchto vah jsou voleny náhodně při vytvoření sítě.



Obr. 2. Struktura neuronové sítě
topologie sítě 2-3-1

Při výpočtu dopředného průchodu se postupuje dle schématu (viz. Obr. 3.). Do první, sumační, části neuronu vstupují hodnoty z neuronů $x_1 - x_n$ z předchozí vrstvy vynásobené vahami $w_1 - w_n$ a offset hodnota *bias*. Hodnota ze sumační části následně vstupuje do aktivační funkce f v druhé části neuronu a z té vystupuje hodnota neuronu y . [3]



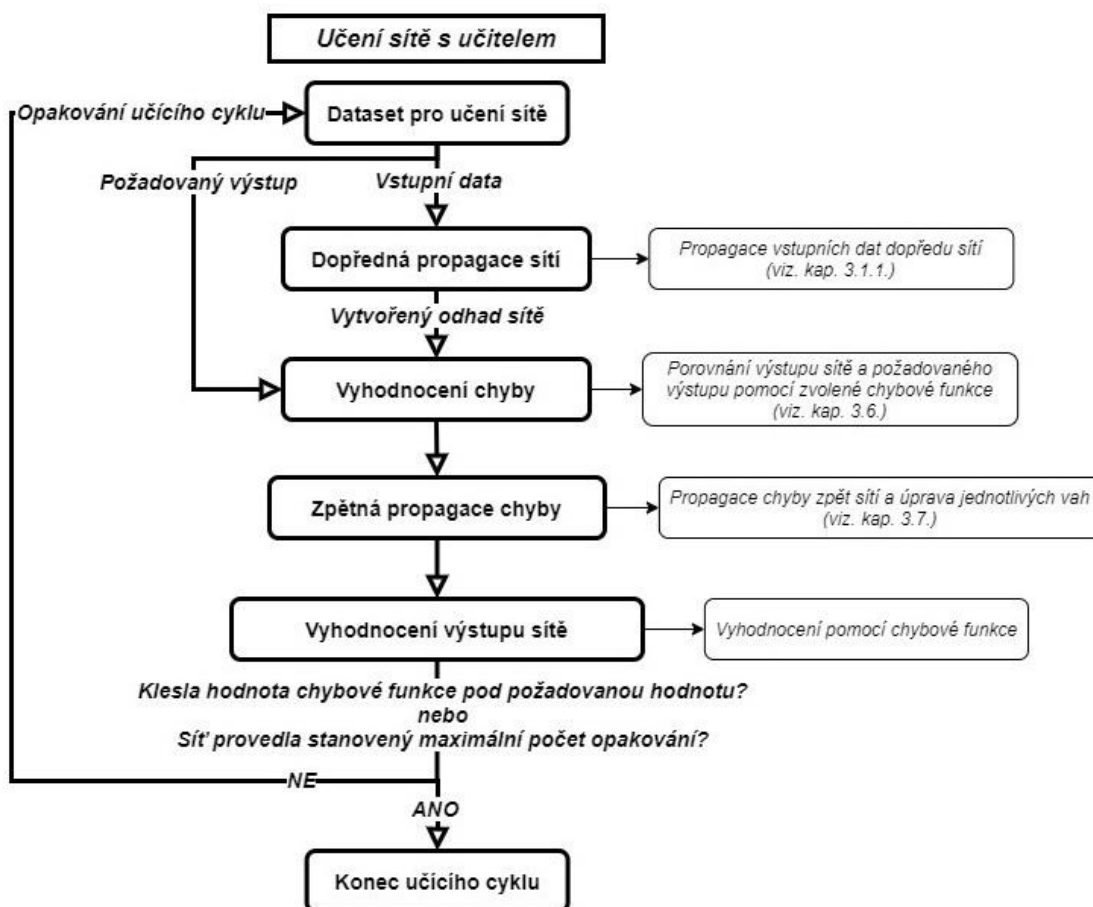
Obr. 3. Struktura neuronu

4.2. Učení sítě

Po vytvoření má síť pouze náhodně zvolené proměnné a na zadaná vstupní data bude vytvářet víceméně náhodné výstupy. Je potřeba ji nejprve „naučit“ rozpoznávat požadované vzory. Samotné učení probíhá iterativní úpravou jednotlivých vah sítě. Učení sítě probíhá na datech s vlastnostmi, které požadujeme, aby byla síť schopna postihnout po naučení. Učení sítě může probíhat dvěma metodami, volenými dle dostupných dat, a to s učitelem, pokud jsou k dispozici označená data, a bez učitele, která ovšem vyžaduje složitější algoritmy zpracování dat. V práci bylo využito pouze učení s učitelem. [4]

4.2.1. Učení s učitelem

Metoda učení s učitelem byla zvolena při vypracování této práce. Jde o metodu, kdy je síť klasifikována na základě jejího odhadu pro daný vstup a požadovaného výstupu. Pro použití metody s učitelem jsou tedy zapotřebí předpřipravená data s přiřazenými hodnotami pro vstup a výstup, což musí být provedeno ručně. Chyba určená učitelem je poté použita pro úpravu sítě a to tak, že jednotlivé váhy jsou při zpětném průchodu sítě upravovány na základě této získané chyby metodou zpětné propagace (*backpropagation*). Síť takto učíme na námi přímo požadovaný výstup s cílem, aby jej replikovala, avšak dostatečně obecně, aby ji bylo možno aplikovat i na neznámá data. [4]

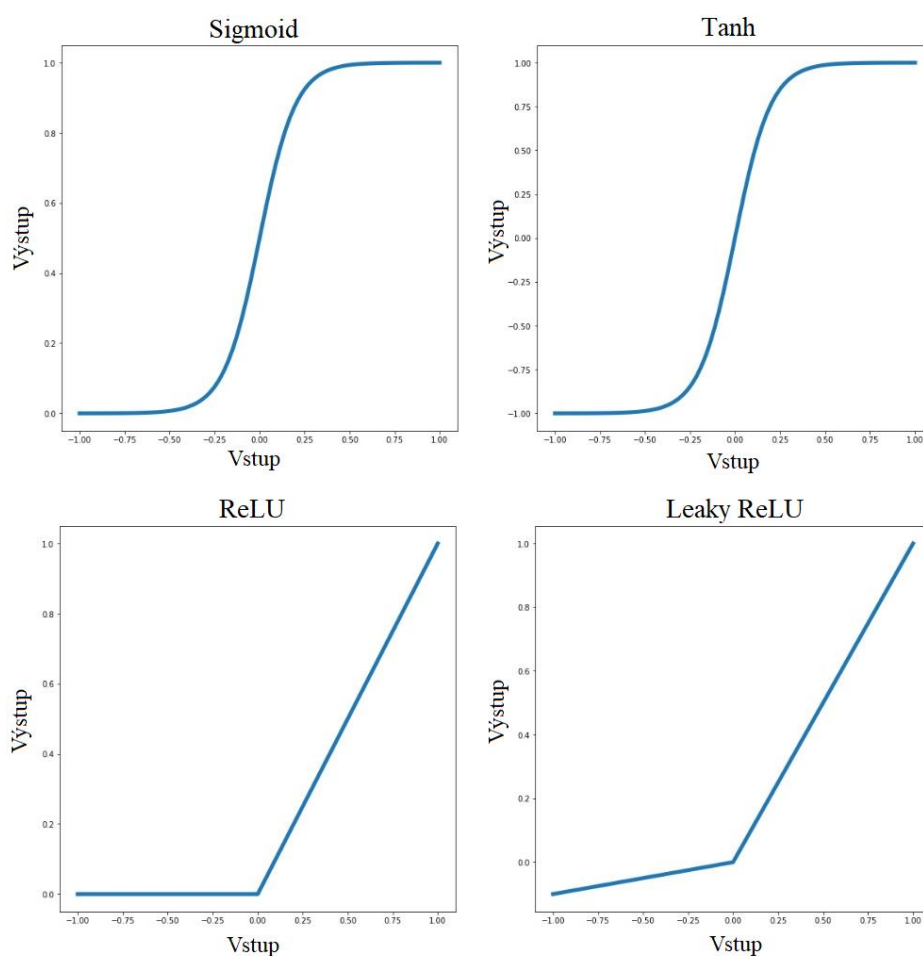


Obr. 4. Struktura učení s učitelem

4.3. Aktivační funkce

Aktivační funkce reprezentuje vnitřní strukturu neuronu, jenž na určitou intenzitu vstupního signálu odpovídá svým výstupem. Při volbě aktivační funkce je potřeba brát v potaz její rozsah při přípravě dat pro učení. Pro práci s algoritmy zpětného šíření je vhodné pracovat pouze s elektrickým modelem neuronu nebo aditivními aktivacemi Grossbergova teorému, jelikož multiplikativní aktivace nejsou jednoduše zpracovatelné a mohou způsobovat komplikace. [5]

Možnými funkcemi jsou například Sigmoid, Tanh, ReLU, Leaky ReLU a další, případně jejich možné kombinace pro různé vrstvy sítě. Průběhy jednotlivých funkcí v závislosti na vstupních datech jsou vykresleny níže (viz. Obr. 5.).



Obr. 5. Průběhy Aktivačních funkcí

4.4. Chybové funkce

Chybová funkce slouží k ohodnocení predikce sítě na základě daných vstupů a požadovaného výstupu a pro následnou úpravu dat. Pokud není síť chybně nastavena, tak je její průběh, při postupném učení sítě, klesající. Výběr chybové funkce záleží na datech, na které bude aplikována, výstupy jednotlivých funkcí ovšem nelze přímo srovnávat, jelikož hodnoty chybové funkce jsou pro každou variantu různé. Možných chybových funkcí je mnoho, proto bude zmíněno jen pár vybraných, které byly používány a testovány. [6]

4.4.1. MSE

Nejčastěji používaná funkce, představuje čtverec vzdálenosti mezi požadovanou a predikovanou hodnotou. Nevýhodou může být, pokud se v datech vyskytují náhodné shluky vzdálenější od požadovaného průběhu, proložení jimi bude silně negativně ovlivněno, jelikož je umocněno se čtvercem.

$$MSE = (y - y_p)^2 \quad (1)$$

4.4.2. MAE

MAE je absolutní vzdálenost mezi požadovanou a predikovanou hodnotou. Netrestá tolik vzdálené body, což je výhoda, pokud s v datech častěji vyskytují náhodné vzdálené body, ale zároveň má větší vliv na blízké body, na rozdíl od MSE, což není příliš žádoucí, jelikož, velikost kroku neklesá s tím, jak se funkce blíží k minimu.

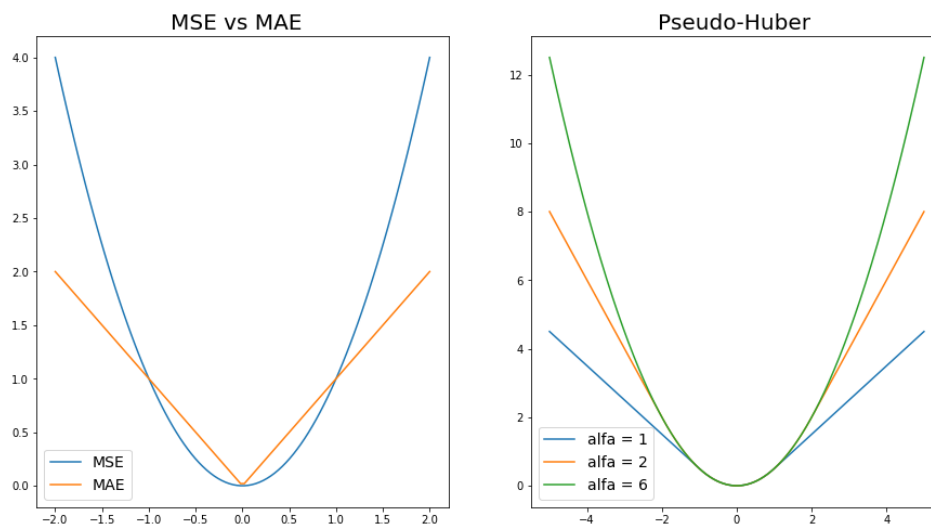
$$MAE = |y - y_p| \quad (2)$$

4.4.3. Pseudo-Huber Loss

Pseudo-Huber Loss kombinací dvou předchozích funkcí a využívá jejich pozitivních vlastností. Pro vzdálenost požadované a predikované hodnoty $\leq \delta$ se chová jako MSE, a pro vzdálenost $> \delta$ jako MAE, kdy δ (*delta*) je volený parametr určující přechodný bod a zároveň sklon MAE.

$$\frac{1}{2} \cdot (y - y_p)^2 \quad ; pro |y - y_p| \leq \delta \quad (3)$$

$$\delta |y - y_p| - \frac{1}{2} \delta^2 \quad ; pro |y - y_p| > \delta \quad (4)$$

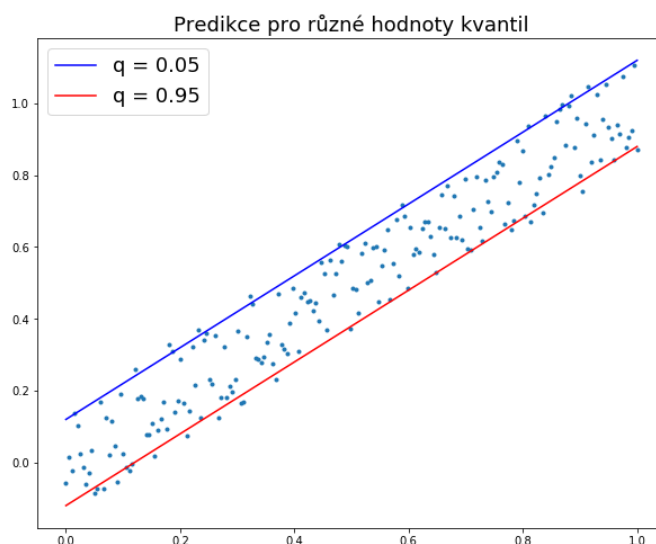


Obr. 6. Průběhy křivek chybových funkcí

4.4.4. Quantile Loss

Quantile Loss se využívá, pokud je potřeba znát pravděpodobnostní prostor, do něž budou data spadat. Je aplikována k predikci a pomáhá určit prostor, do kterého by měly hodnoty spadat. Kvantil poté značí procentní hodnotu, pod níž spadají vyhodnocované body. [7]

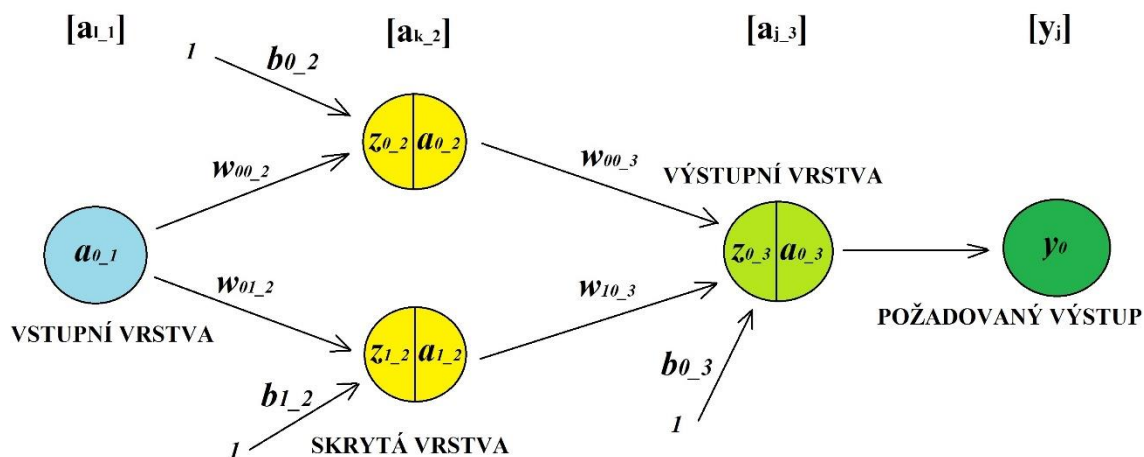
$$L(y_p, y) = \max[q \cdot (y_p - y); (q - 1) \cdot (y_p - y)] \quad (5)$$



Obr. 7. Vliv velikosti kvantilu na proložení

4.5. Backpropagation

Zpětná propagace neboli backpropagation, počítá gradient funkce a upravuje jednotlivé hodnoty, aby se docílilo dosažení, optimálně globálního, minima funkce. Jelikož každá vstupní hodnota prochází vícero vrstvami aktivačních a sumačních funkcí, je využíváno řetězového pravidla (*chain rule*) k provedení chyby zpět sítě a úpravy jednotlivých požadovaných vah. Zpětná propagace bude provedena ilustračně na jednoduché síti, pro jednodušší pochopení (viz. Obr. 8).



Obr. 8. Schéma pro zpětnou propagaci

$$C_o = \sum (a_j^3 - y_o)^2 \quad (6)$$

$$a_j^3 = \sigma(z_j^3) \quad (7)$$

$$z_j^3 = (w_{00}^3 \cdot a_0^2 + w_{10}^3 \cdot a_1^2 + b_0^3) \quad (8)$$

Kde C_o je chyba mezi výstupem ze sítě a požadovaným výstupem, z je suma vstupů z předchozí vrstvy, σ je aktivační funkce, a je výstup z neuronu po aktivační funkci y je požadovaný výstup, w jsou váhy jednotlivých vrstev a b jsou bias hodnoty každého neuronu. Aplikací rovnice (9) tedy získáme přírůstek váhy pro rovnici (10).

$$\frac{dC_o}{dw} = \frac{dz}{dw} \cdot \frac{da}{dz} \cdot \frac{dC_o}{da} \quad (9)$$

Pomocí aplikace řetězového pravidla a zpětnou propagací je získána úprava váhy dle gradientu (dw). Ta je násobena učícím koeficientem (*learn rate*) a přičtena k původní hodnotě váhy (w_o).

$$w_o = w_o + learning_rate \cdot dw \quad (10)$$

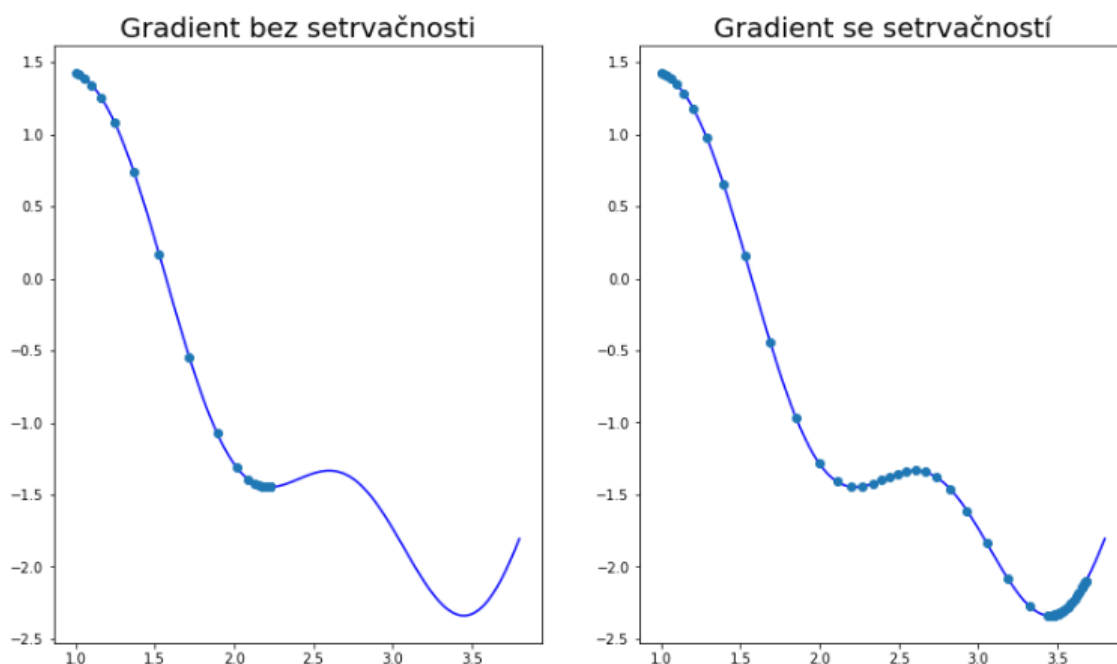
Při zpětné propagaci do dalších vrstev je potřeba brát v potaz, že C_0 je suma přes všechny prvky, které byly při dopředném průchodu ovlivněny. Pokud bychom tedy chtěli počítat dopředu pro vstupní vrstvy, bude chyba C_0 sumou ze všech prvků následující vrstvy a je určena z modifikace rovnice (9) v rovnici (11), kde da na levé straně rovnice je výstup ze skryté vrstvy a_x^2 a da na pravé straně rovnice, výstup z výstupní vrstvy a_x^3 .

$$\frac{dC_0}{da} = \frac{dz}{dw} \cdot \frac{da}{dz} \cdot \frac{dC_0}{da} \quad (11)$$

Je tedy potřeba postupovat sítí odzadu a průběžné požadované výstupy z buněk ukládat, jinak s hloubkou sítě roste náročnost výpočtu exponenciálně.

Princip učení neuronové sítě pomocí zpětné propagace je tedy v inkrementálních úpravách jednotlivých vah, na základě chyby na výstupu. Proto rovněž platí, že čím větší objem dat je možno sítí poskytnout tím lepší výsledek bude schopna doručit postupným učením se. [8]

K vahám je možno přidat i setrvačnost pro snazší překonávání malých lokálních minim a nezasekávání se, ovšem za cenu pomalejšího posuvu po gradientu a možného dokmitávání do požadovaného minima (viz. Obr. 9.).



Obr. 9. Překonání plytkého lokálního minima gradientem se setrvačností

4.6. Learning rate

Koeficient učení neboli *Learning rate* udává, jak rychle se budeme snažit síť učit. Udává s jak velkým krokem se chceme po gradientu pohybovat. Volba správné velikosti je náročná a může výrazně ovlivnit kvalitu naučené sítě i rychlost jejího učení, jelikož pokud bude koeficient zvolen příliš velký, bude průběh učení náhodně skákat po křivce gradientu a chyba se může spíše zvětšovat nežli klesat, naopak pokud bude příliš malý, tak bude učení probíhat příliš pomalu, nebo se může lehce zaseknout v lokálním minimu.

Analytický výpočet optimálního koeficientu není možný, a tak je potřeba jej odhadnout, nebo určit experimentálně. Dobrou výchozí hodnotou mohou být hodnoty 0.1 nebo 0.01 záleží ovšem opět na řešené síti. Pomocníkem může být vykreslování chyby, nebo vlivu různých velikostí koeficientu na síť v závislosti na počtu uplynulých epoch.

Možnými variantami pokročilejších úprav koeficientu učení jsou například přidání momentu setrvačnosti (viz. Obr. 9.), což má za následek pohyb váhy ve směru, v němž se předtím déle pohybovala, pomáhá urychlit proces učení v závislosti na počtu epoch a snáze překonává lokální minima a zakřivení gradientu. Dále například použití křivky, nebo cyklu po jehož hodnotách se koeficient pohybuje, nebo adaptivního algoritmu, jenž upravuje velikost koeficientu v závislosti na monitorování výkonnosti sítě na datasetu.

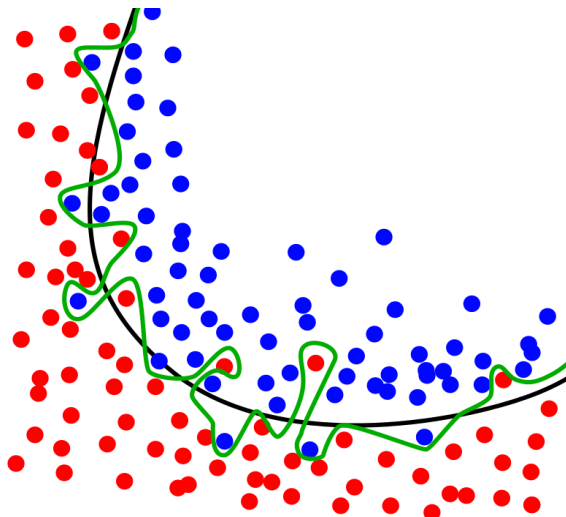
Dalším faktorem ke zvážení může být pokles vlivu učení v závislosti na tom, jak zpětná propagace postupuje sítí zprava doleva. To má za následek, že váhy blíže ke vstupním neuronům se učí pomaleji než váhy u neuronů výstupních. To je možno regulovat rozdílnou hodnotou koeficientu učení pro jednotlivé vrstvy sítě.

Ačkoli ne přímo související s koeficientem učení, dalším prvkem hodným pozornosti je záměrný mírný úbytek hodnot vah neboli *weight decay*, jenž má za úkol regulovat konstantní růst vah při učení sítě. Pokud je síť dlouho trénována, dochází k neustálému nárůstu vah a ačkoli se tak systém stává přesnější, stávala se také náchylnější na šum, a na drobné rozdíly v datech by mohl reagovat nepřiměřeně prudce. Proto je snaha normalizovat data směrem k nižším hodnotám. Druhou potenciální výhodou aplikace tohoto prvku je postupné vymizení vah, které na systém nemají vliv a jsou přebytečné. Pokud vymizí předem určené množství vstupních vah některého z neuronů, je možné jej odstranit jako přebytečný a urychlit si tímto výpočet, (viz. kap. 4.8.1 *Prořezávání sítě*). Nevýhodou je, že malé hodnoty vah vnášejí do systému dodatečnou přesnost a jejich vymizením, nebo odstraněním se může klesnout přesnost celého systému. Aplikace algoritmu prořezávání sítě je tedy vhodná při snaze určit přibližnou topologii, jenž má výsledná síť mít, ale její použití je při finálním učení již nežádoucí. [9]

4.7. Příprava dat

Vhodná příprava dat pro učení sítě je velmi kritickým krokem pro výslednou kvalitu naučené sítě. Síti by měly být poskytnuty, pokud možno, všechny relevantní vstupní proměnné, pro analýzu záznamu vibrací ložiska by byly relevantními proměnnými k samotnému záznamu například zátěž a otáčky při nich bylo měření prováděno. Pokud se v datech vyskytuje časové zpoždění je vhodné jej odstranit. Šum v datech je možno nechat, pokud není příliš výrazný, jelikož napomáhá zabránit přetrénování sítě.

Před samotným učením je vhodné si data rozdělit na dva soubory, a to na učicí a kontrolní set, pokud je to možno, kdy doporučený orientační poměr je 80:20 ve prospěch učicího datasetu. Pro učení sítě bude využit pouze učicí set a na kontrolní sadě se ověří výstup sítě. Tento přístup se používá z důvodu prevence přetrénování („*overfitting*“) sítě (viz. Obr. 10.), což má za následek, že se síť začne učit na nuance daného vzorku na úkor hlavního vzoru a při aplikaci na ostatní vzorky tak bude její chování výrazně horší.



Obr. 10. Výstup sítě naučené příliš přesně na vzorek dat

Prevenčí je kontrola výstupní chyby, která, pokud bude dlouho příliš malá, může to znamenat, že jsou data síti již známá a začíná se učit její drobné odchylky a je tedy potřeba přejít na jiný vzorek. Druhou variantou je právě využití kontrolního setu, porovnávání výsledků a sledování výstupní chyby z obou setů. V případě, že hodnoty pro kontrolní set začnou růst, značí to, že síť se začíná přeučovat z obecných vzorů na konkrétní vzory daného setu, což je nežádoucí a je potřeba trénink na daných vzorcích ukončit.

Data jednotlivých setů je dále možno rozdělit do jednotlivých skupinek (*batches*), které vždy projdou sítí a budou vyhodnocena jejich chyba, ale ke zpětné propagaci dojde až na konci každé skupinky. Doporučený objem každé skupinky je přibližně mezi 1-32 prvky. Dojde tak ke zrychlení učení a menší citlivosti na případný šum v datech, klesá tím ale rovněž případná přesnost sítě, jelikož tak dochází k jisté filtraci dat.

4.8. Vyhodnocování dat

Způsobů vyhodnocení signálů je opět vícero. Zde bude zmíněno pár základních, které byly při vytváření práce testovány s cílem získání sítě nejlépe reagující a vyhodnocující stav součástí. Metody analýzy lze rozdělit do dvou základních skupin. Skupiny je možno dále dělit, zde jsou však zmíněny pouze základní zvažované varianty.

- *V časové oblasti* – signál v časové oblasti je záznam naměřený přímo snímačem, který představuje průběh vibrací v závislosti na čase. Umožňuje analyzovat některé typy poruch snáze než analýza ve frekvenční oblasti. Do ní spadá možný způsob analýzy pomocí.
 - *Celková úroveň vibrací* – jde pravděpodobně o nejjednodušší způsob vyhodnocení ze signálu v časové oblasti srovnáváním celkové amplitudy vibrací. Metoda může být vhodná pro vysoko-rychlostí rotační součásti. Má však výrazná omezení.
- *Ve frekvenční oblasti* – signál je z časové oblasti převeden do frekvenční pomocí Fourierovy transformace. Výstupem je vykreslení frekvencí a amplitudy signálu. Většina analýz vibrací se provádí právě ve frekvenčním spektru. Do ní spadá možný způsob analýzy pomocí.
 - *Spektrální analýza vibrací* – spočívá v převedení signálu do frekvenční oblasti a jsou vyhodnocovány potenciální výrazné frekvence, nebo změny v celkovém signálu. Pokud se v signálu vyskytuje vrchol, lze předpokládat, že se ve v měřené součásti nachází zdroj vibrací. Při analýze může být i podstatným vstupem rychlost otáčení, což má za následek posuvy frekvencí úměrně dané frekvenci.
 - *Diskrétní monitorování frekvencí* – spočívá ve znalosti, které komponenty mají být monitorovány a které frekvence jim přísluší. Z frekvenčního spektra jsou poté tyto frekvence extrahovány, pro vyhodnocení.

Při zpracování dat ve frekvenčním spektru je možné se zaměřit pouze na spektra v nichž je analyzovaná porucha očekávána. Obecně lze tyto spektra rozdělit následovně.

- *Méně než 1x otáčkové frekvence* – výskyt nestability oleje u kluzných ložisek
- *1x až 10x otáčkové frekvence* – výskyt závad jako uvolnění, nevývaha, nesouosost
- *Více než 10x otáčkové frekvence* – výskyt závad valivých elementů ložisek

Primární zájem této práce je směřován na analýzu závady valivých elementů ložisek. Stěžejní analýza tedy bude nad 10-ti násobkem otáčkové frekvence. Pokud by se jednalo diskretní monitorování frekvencí, přibližné frekvence závady lze vypočítat pro jednotlivé elementy ložiska zvlášť, je ovšem vyžadována znalost typu a rozměrů ložiska.

- *BPFO* – frekvence při poškození vnějšího kroužku, rovnice (12)
- *BPFI* – frekvence při poškození vnitřního kroužku, rovnice (13)
- *BSF* – frekvence při poškození valivých elementů, rovnice (14)
- *FTF* – frekvence při poškození klece, rovnice (15)

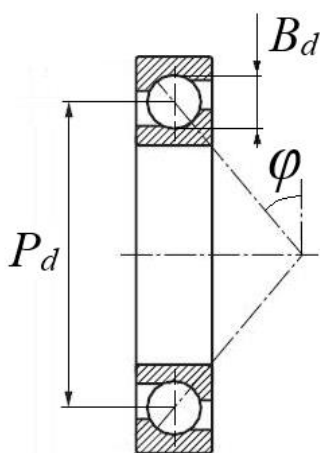
$$BPFO = \frac{n}{2} \cdot \frac{rpm}{60} \cdot \left(1 - \frac{B_d}{P_d} \cdot \cos \varphi\right) \quad (12)$$

$$BPFI = \frac{n}{2} \cdot \frac{rpm}{60} \cdot \left(1 + \frac{B_d}{P_d} \cdot \cos \varphi\right) \quad (13)$$

$$BSF = \frac{P_d}{2 \cdot B_d} \cdot \frac{rpm}{60} \cdot \left(1 - \left(\frac{B_d}{P_d}\right)^2 \cdot \cos^2 \varphi\right) \quad (14)$$

$$FTF = \frac{P_d}{2 \cdot B_d} \cdot \frac{rpm}{60} \cdot \left(1 - \frac{B_d}{P_d} \cdot \cos \varphi\right) \quad (15)$$

Kde n je počet valivých elementů [-], B_d je průměr valivého ložiska [mm], P_d je roztečný průměr ložiska [mm], φ je kontaktní úhel [deg] a rpm jsou otáčky rotoru [ot/min] (viz. Obr. 11.). [10][11]



Obr. 11. Náčrt ložiska

4.9. Volba sítě

Různé struktury neuronových sítí se skládají ze stejných základních prvků, neuronů. Podle vzájemného uspořádání těchto prvků a jejich vlastností se rozlišují různé typy sítí. Základní prvky popisu struktur sítí jsou:

- *Topologie* – počet vrstev sítě a neuronů obsažených v každé vrstvě (např.: 2-4-2)
- *Architektura* – zvolený typ sítě a funkce neuronů (např.: *dopředná*, *rekurentní*)

Volba architektury sítě udává, jaká bude vnitřní struktura vzájemných propojení sítě a jaké budou použity neurony.

- *Dopředná* – (např.: *MPL – Multi Layered Perceptron*) kde propojení neuronů jsou pouze do následující vrstvy, a to buď v plném propojení, nebo pouze částečném a výstup je vždy pouze reakcí na aktuální vstupní data. Jedná se o jednu z nejběžnějších variant architektur. Pro učení sítě využívá algoritmu zpětné propagace (*Backpropagation*).
- *Rekurentní* – (např.: *RNN – Recurent Neural Network*) kde propojení neuronů je i zpětnovazební, buď do stejné vrstvy, nebo do předchozích vrstev, síť tak má určitou paměť a výstup není pouze reakcí a aktuální vstup, ale i na vstupy předchozí.
- *Konvoluční* – (např.: *CNN – Convolutional Neural Network*) propojení neuronů je pouze do následující vrstvy, neurony provádí konvoluční filtraci pro zvýraznění vzorů, například hran ve 2D formátu dat.

Při volbě architektury sítě byla zvažovanou problematikou neexistence paměti u dopředné sítě, což by bylo možno řešit použitím varianty rekurentní sítě, jejíž neurony mají výstup, který je zároveň jejich vstupem. Je ovšem potřeba opatrně zvolit váhy těchto zpětných vazeb které, pokud budou váhy příliš malé, tak dojde k rychlému vymizení informace, naopak pokud budou rekurentní váhy příliš velké, dojde k prudkému nárůstu hodnoty na vstupu neuronu, jejímu přetečení, a síť pak nebude schopna fungovat. Je tedy vhodné implementovat algoritmus regulující vstupní a výstupní hodnoty z rekurentního neuronu. Jednou z nejjednodušších variant rekurentní sítě je síť o jedné skryté rekurentní vrstvě, jenž má náhodně zvolené váhy vstupů a rekurentních propojení, které se již dále netrénují a jedné skryté dopředné vrstvě, jenž vyhodnocuje výstup z rekurentní vrstvy a je trénována. Jediným požadavkem na takto konstruovanou síť je násobně větší počet neuronů ve skryté vrstvě, než je počet neuronů vstupní vrstvy. Bylo zvažováno i použití architektury s rekurentní vrstvou a její aplikace byla testována, ale nakonec od ní bylo upuštěno ve prospěch dopředné architektury.

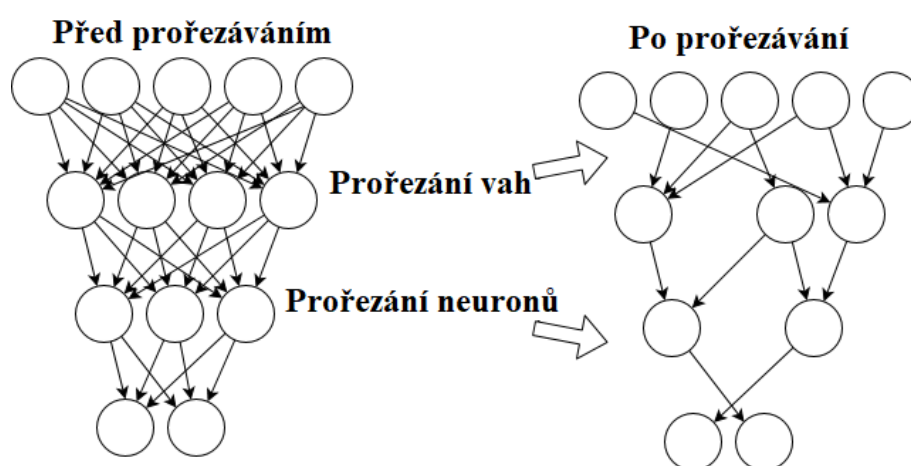
Pro práci byla tedy nakonec zvolena varianta dopředné sítě s plným propojením vrstev v architektuře MPL (*Multi Layered Preceptron*).

Volba topologie sítě závisí na požadované analýze. Pro jednoduchá, lineárně separovatelná data stačí pouze perceptronová síť o vstupní a výstupní vrstvě. Pokud je v datech nelinearita, je možno to řešit přidáním vrstvy. Přidáváním vrstev dáváme síti k dispozici další dimenze, ve kterých může prostor s daty různě deformovat. Topologie sítě byla určena experimentálně, a to cyklickým testováním v automatizované smyčce pro různé velikosti vrstev ve variantách s topologií $300-x-1$ a $300-x-y-1$, varianta sítě s 3-mi skrytými vrstvami byla zavržena v důsledku velké výpočetní a tedy i časovou náročností na otestování těchto variant.

Snahou byl nalézt suboptimální přibližnou topologii, která by poskytovala uspokojivé výsledky a z prováděných testů odhadnout v jakém rozsahu velikostí jednotlivých skrytých vrstev se může topologie pohybovat. [12][13]

4.9.1. Prořezávání sítě

Prořezávání sítě (*pruning*), je metoda odstraňování přebytečných vah, případně neuronů. Využívá se buď jako doplňující varianta k „záměrnému postupnému úbytku hodnot vah“, (viz. kap. 4.6. *Learning rate*), kde má za úkol odstraňovat váhy, jejich hodnota se blíží nule a stanou se tak zbytečnými, nebo přímo jako doplňující metoda tvorby sítě, kdy v tomto případě se síť záměrně navrhne větší, než je dle odhadu potřeba a v průběhu učení se odstraňují váhy a neurony s nejmenšími posuny, nebo vlivem (viz. Obr. 12.). Odstranění prvků touto metodou je trvalé a zmenší tak velikost sítě a tím i výpočetní náročnost. Velikost prořezání může být dána jako požadavek na procentní redukci, která probíhá kontinuálně během učení, dokud jí není dosaženo, nebo pouze jako redukce vah jdoucích k nulové hodnotě.



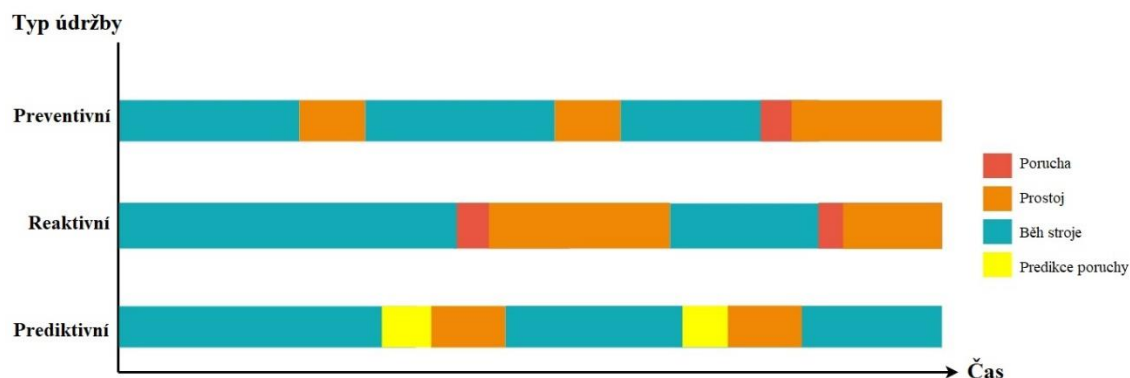
Obr. 12. Výstup po prořezávání sítě

Záměrný úbytek hodnot (*weight decay*) je algoritmus, který v určitých intervalech, obvykle nikoli ihned po startu a rovněž ne před finálním doučením sítě, redukuje velikost jednotlivých vah sítě, jelikož během učení dochází k jejich postupnému nárůstu. Velikost úbytku a je potřeba zvolit rozumně, aby příliš nekolidoval s postupem v učení sítě samotné. Prořezávání bylo používáno jako doplněk k učení sítě při testování různých topologií pro teoretickou optimalizaci rozměrů jednotlivých skrytých vrstev odstraňováním nepotřebných vah.

Cílem bylo z vytvořeného odhadu topologie vytvořit síť s redukováným počtem prvků, bez výrazné ztráty funkčnosti sítě a tím snížit výpočetní náročnost na učení a doučování sítě samotné. [14]

4.10. Odhad zbývajícího života

Odhad zbývajícího života neboli RUL (*remaining usefull life*), slouží k odhadu času, po který bude moci stroj pracovat, než bude vyžadováno provedení údržby, nebo opravy. Odhad zbývajícího času je důležitý pro plánování preventivní údržby a předejití neplánovaných odstávek z důvodu závady, kdy neplánovaná odstávka je časově výrazně náročnější na čas, kdy stroj není v provozu, nežli plánovaná údržba (viz. Obr. 13.).



Obr. 13. Srovnání preventivní, reaktivní a prediktivní údržby dle na časové náročnosti

Preventivní údržba spoléhá na pevně dané intervaly pro výměnu součástí, i když součást může být stále v dobrém stavu. Ačkoli takovéto řešení postihne většinu problému před jejich vznikem, má za následek plýtvání materiálem a nemůže stále zaručit bezchybný chod v případě vady na vyměněném dílu a následnou neočekávanou poruchu a s ní spojený prostoje v běhu stroje.

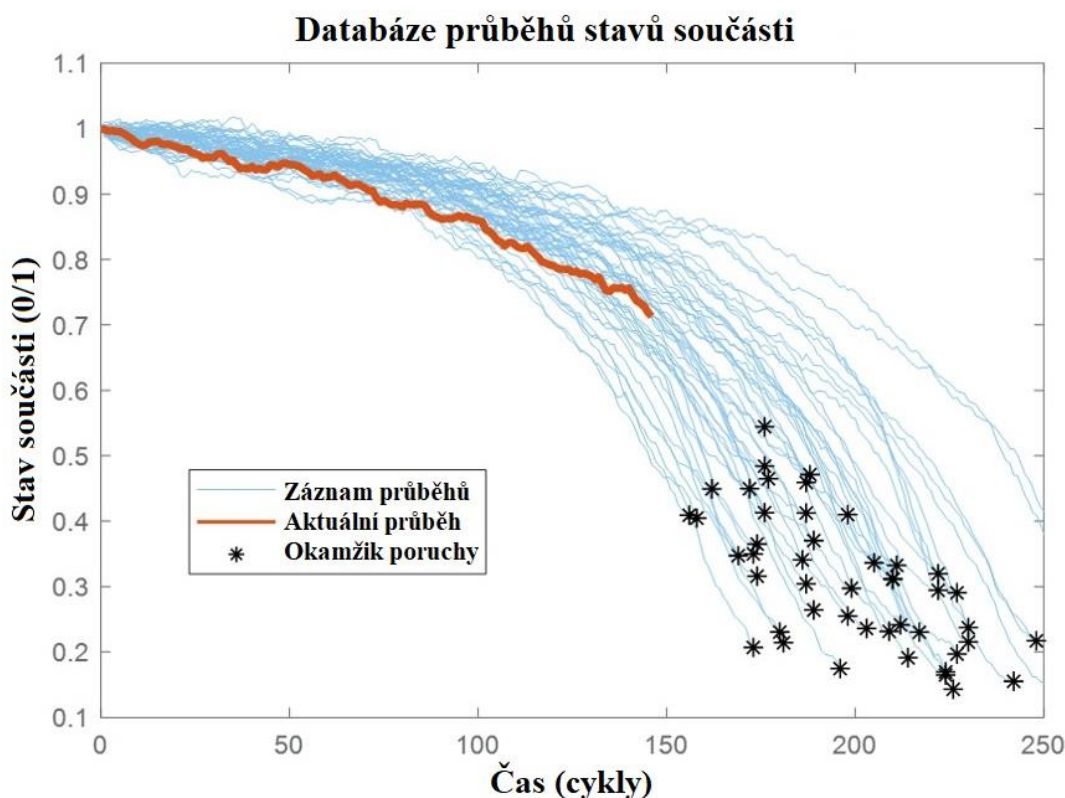
Reaktivní údržba využívá maximální životnosti dílu a výměna je provedena až v okamžiku, kdy dojde k jeho poruše. Ačkoli je takto dosaženo delších časů mezi jednotlivými výměnami, jedná se o řešení dosti neefektivní, jelikož prostoje po každé poruše jsou velmi zdlouhavé a rovněž poškozená součást při své poruše může poškodit i součásti ve svém okolí a tím výrazně zvýšit náklady a čas na potřebnou opravu.

Prediktivní údržba, se na rozdíl od dvou předchozích metod, snaží předpovídat stav součástí a na jeho základě určit čas zbývající do poruchy, aby bylo možno plánovat údržbu stroje, bez prostoje způsobených nenadálou poruchou a zároveň využít celého života součástí, čímž dojde k prodloužení časů mezi jednotlivými opravami a sníží se poměr prostoje stroje vůči jeho chodu. Použití této metody vyžaduje monitorování běhu stroje a systém vyhodnocování dat. Ani použití této metody není neomylné a stále může dojít k poruše, kterou systém nebude schopen vyhodnotit, nebo poruše dílu, jehož stav nebude systém schopen z dostupných dat vyhodnotit. Oproti předchozím metodám ovšem nabízí nesporné výhody.

Metody pro odhad RUL se liší podle zpracování a dostupnosti dat. Vybrány a testovány byly tři následující metody. [15]

4.10.1. Run-to-Failure metoda

Metoda se zakládá na databázi obsahující průběh stavů dané součásti za jeho života a také bod v němž došlo k poruše součásti. Jednotlivé vzorky pak zachycují degradační proces v závislosti na čase, nebo cyklech. Při odhadování zbývajících času je vstupem do sítě záznam měření ve frekvenčním spektru a výstupem je odhadnutý stav ložiska v daném okamžiku. Průběh degradace je následně porovnáván s touto databází (viz. Obr. 14.) a je vybrán předchozí vzorek z databáze nejlépe odpovídající aktuálnímu průběhu a na jeho základě je odhadnut zbývajících čas dané součásti do její poruchy.



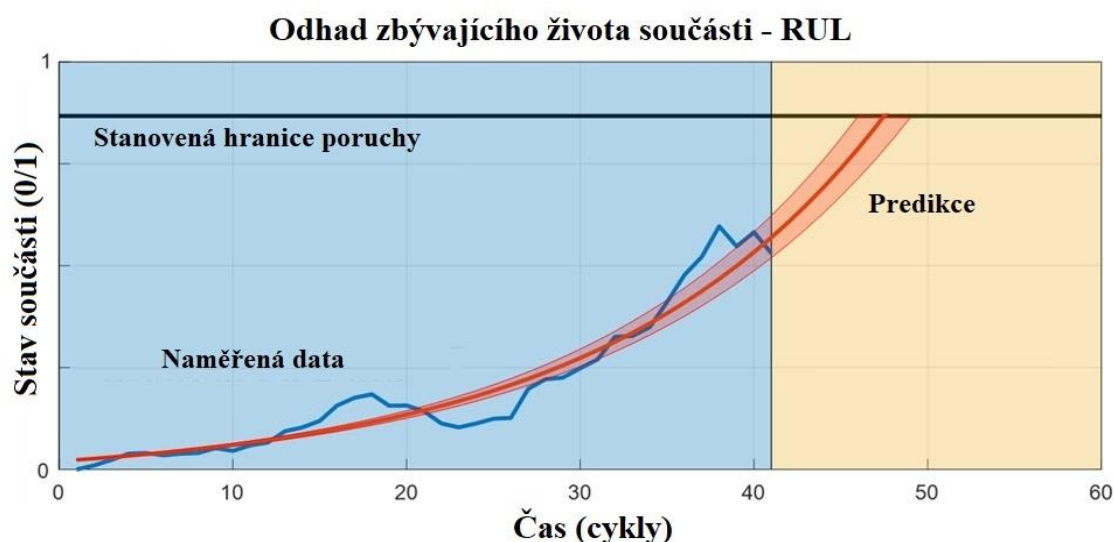
Obr. 14. Vzorek (červená) proložený na databázi poruch

Nevýhodou tohoto řešení je náročnost na velikost databáze, s níž je vzorek porovnáván, a na obdobnost trendu vyskytujících se poruch. Naopak je již v průběhu měření schopen poskytnout relativně uspokojivý odhad zbývajících času.

4.10.2. Threshold metoda

Metoda využívá znalosti předepsané hraniční hodnoty, jež by neměla být překročena. Jednotlivá vyhodnocení stavů v časové závislosti jsou proložena modelem a je vyhodnocováno protnutí s touto hraniční hodnotou a následně je určen RUL. Vstupem je stejně jako u předchozí metody záznam měření ve frekvenčním spektru a výstupem je odhad stavu ložiska, je ovšem vyhodnocován stav na základě překročení dané hranice, buďto aktuálním vzorkem, nebo aproximací na nich vytvořenou (viz. Obr. 15.).

Metoda nevyžaduje žádnou předešlou znalost průběhů poruch na rozdíl od předchozí metody, je ovšem náročná na zvolený aproximační model. Ačkoli jsou data obvykle exponenciální, tak jejich rozptyl může aproximačním algoritmem dělat potíže a některé výsledky, se vlivem rozptylu hodnot z nedokonalé, nebo zašuměné, estimace stavu součásti, mohou být nesmyslné.

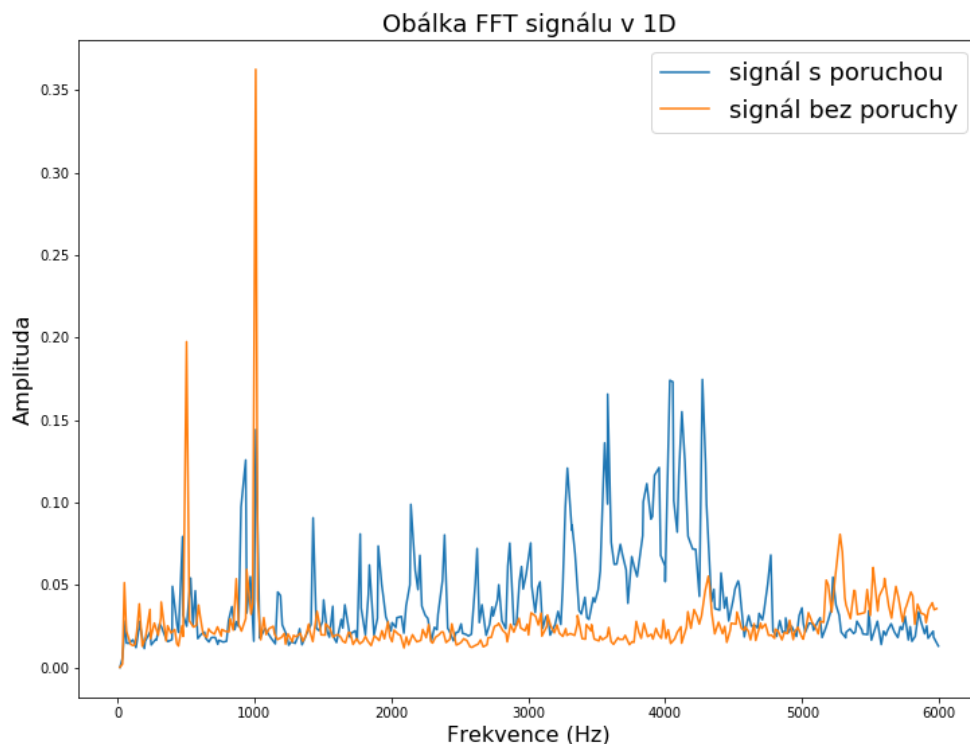


Obr. 15. Threshold metoda a určení RUL

4.10.3. Klasifikátor skupin

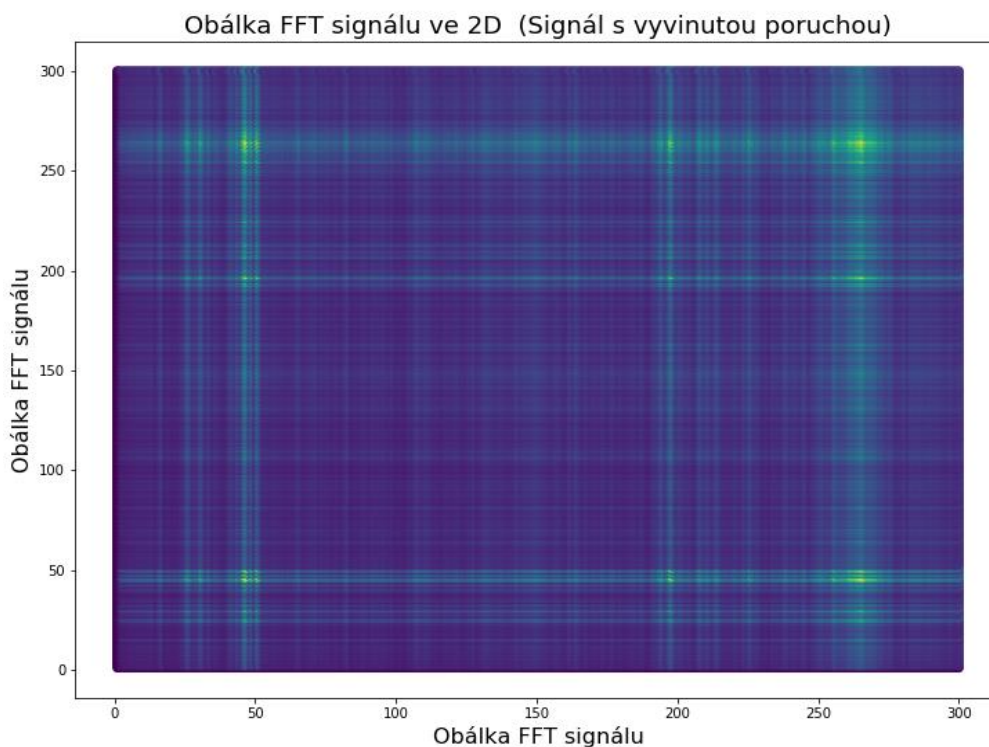
Tato metoda využívá rozdělení dat do skupin s různým rozsahem zbývajících hodin a vytvoření sítě na principu klasifikátoru kdy vstupem je záznam z měření ve frekvenčním spektru, a to buď jakožto 1D formát frekvenčního spektra (viz. Obr. 16.) pro základní strukturu ANN vyhodnocovanou klasickou dopřednou sítí, nebo jako 2D formát frekvenčního spektra, kdy na ose x i y je vytvořená FFT transformace (viz. Obr. 17.), která je vyhodnocována kombinací konvoluční sítě a klasické dopředné ANN. Realizována však byla pouze první varianta, a to vyhodnocení spektra pouze v 1D formátu pomocí dopředné sítě s architekturou MLP. [16]

Nevýhodou této metody je, že přesnost odhadu RUL je pouze na základě množství zvolených skupin (např.: $[0-10h] = 1.sk$ $[10-30h] = 2.sk$ $[30h \text{ a více}] = 3.sk$). S větším počtem skupin a tím přesností, rostla náročnost na učení sítě a její citlivost na data.



Obr. 16. Obálka FFT signálu pro vyhodnocení stavu ložiska
každá z obálek je tvořena 300 body

Při učení byla rovněž použita diskriminační matice, pro úpravu chybových hodnot, dle kterých bude síť upravována. Je výhodnější, aby síť raději vykazovala menší hodnotu RUL, a v případě blížící se poruchy, byla obsluha upozorněna dříve, nežli aby obsluha nebyla upozorněna a došlo k poruše stroje.

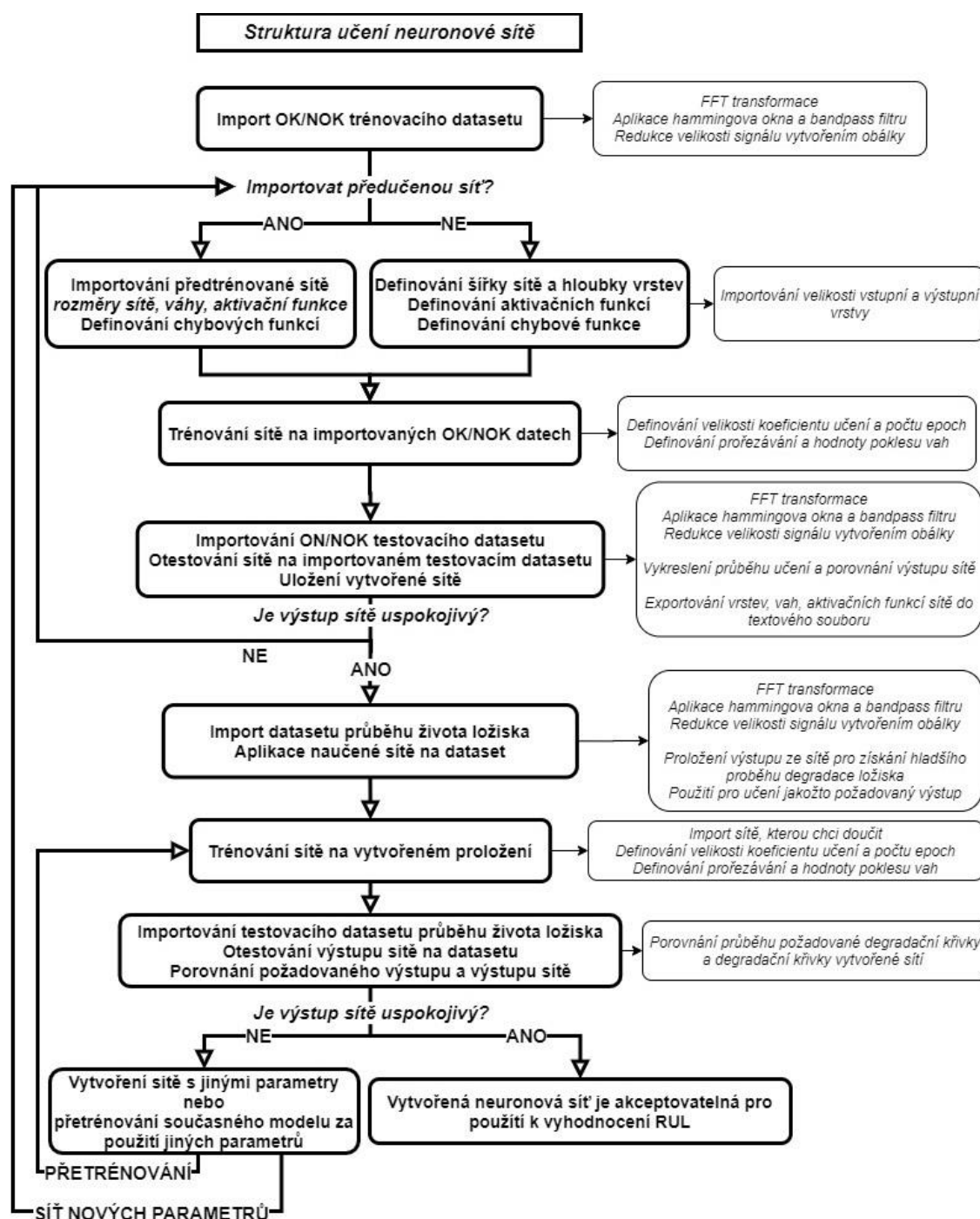


Obr. 17. Transformace 1D FFT na 2D pro konvoluční NN

5. Vypracování tvorby sítě

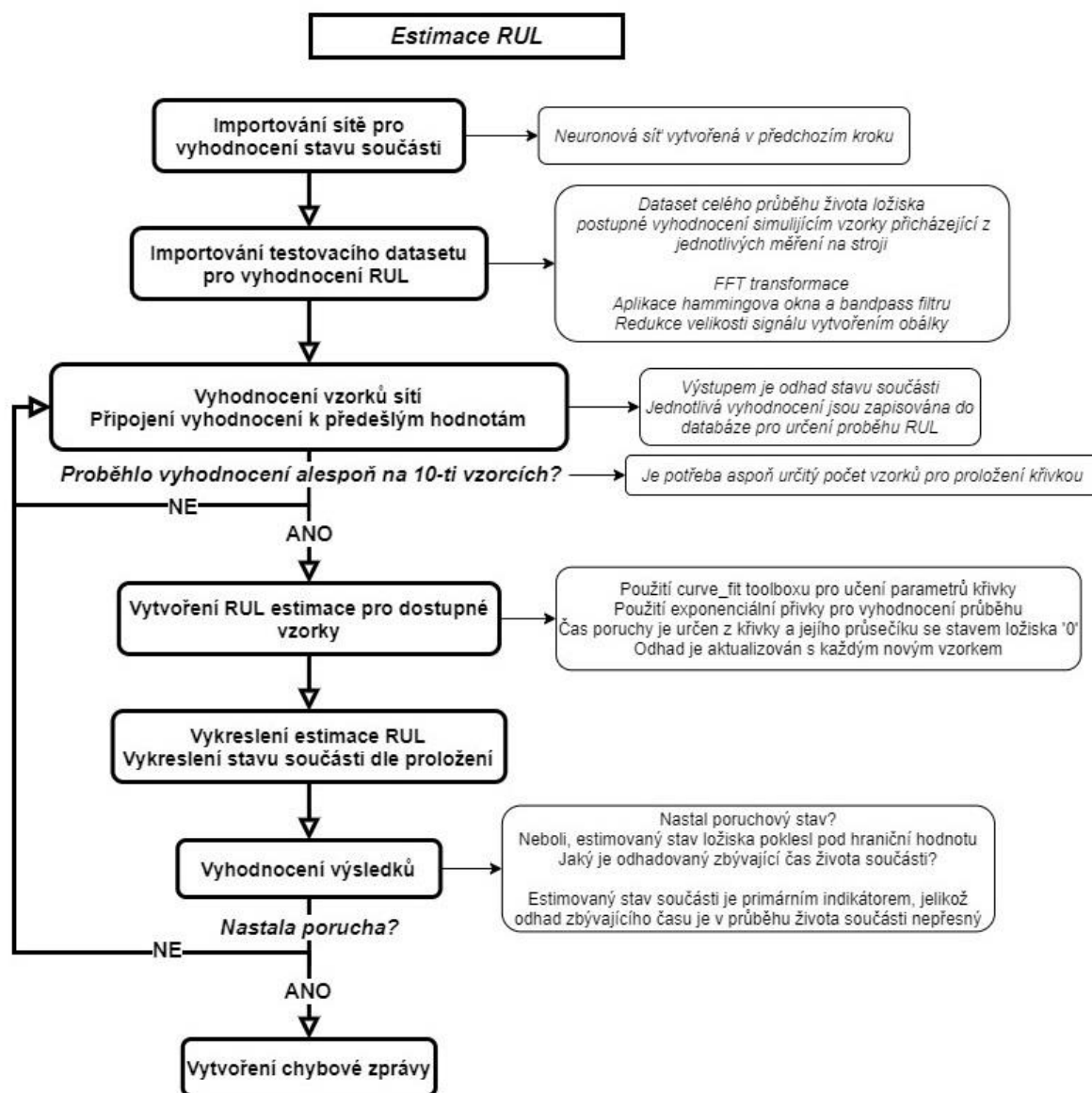
V následující kapitole bude rozebráno vypracování knihovny a algoritmů neuronové sítě, dle rozboru provedeného v přechozí kapitole a vypracování navržené architektury MPL učené s učitelem pomocí algoritmu zpětné propagace.

Zpracování problému bylo, rozděleno do dvou částí, a to na učení neuronové sítě pro vyhodnocování stavu součásti, které je schématicky rozebráno níže (viz. Obr. 18.).



Obr. 18. Struktura učení neuronové sítě

A na následnou estimaci zbývajících života ložiska neboli RUL (*Remaining Usefull Life*) (viz. Obr. 19.) z vyhodnoceného stavu.

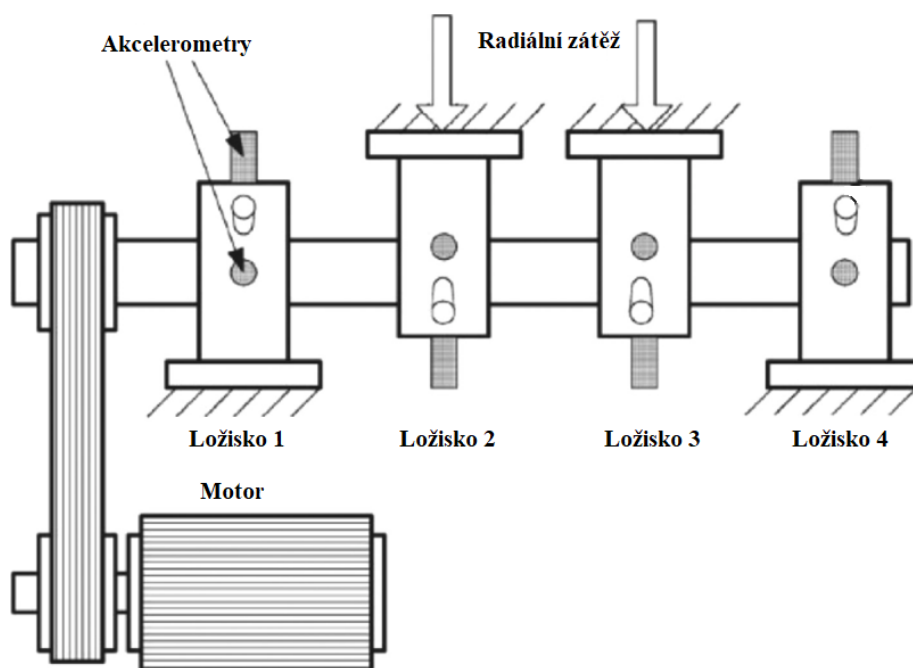


Obr. 19. Struktura odhadu RUL

5.1. Volba a zpracování dat

Za data k učení byl zvolen balíček z měření průběhu života ložisek v testovacím prostředí měřeného za konstantních otáček 2000rpm a pod konstantní zátěží 6000lbs (viz. Obr. 20.). Měření na ložiscích bylo prováděno každých 10 minut po dobu jedné sekundy se vzorkovací frekvencí 20kHz. Z měření poté byly vybrány data ložisek, u nichž se vyskytla porucha a byly následně zpracovány do požadovaných datasetů. Ložiska byla průběžně kontrolována, není ovšem známo, jak byl finální stav vyhodnocen, jestli zadřením některého z ložisek, nebo vyhodnocením poškození při kontrole. [17]

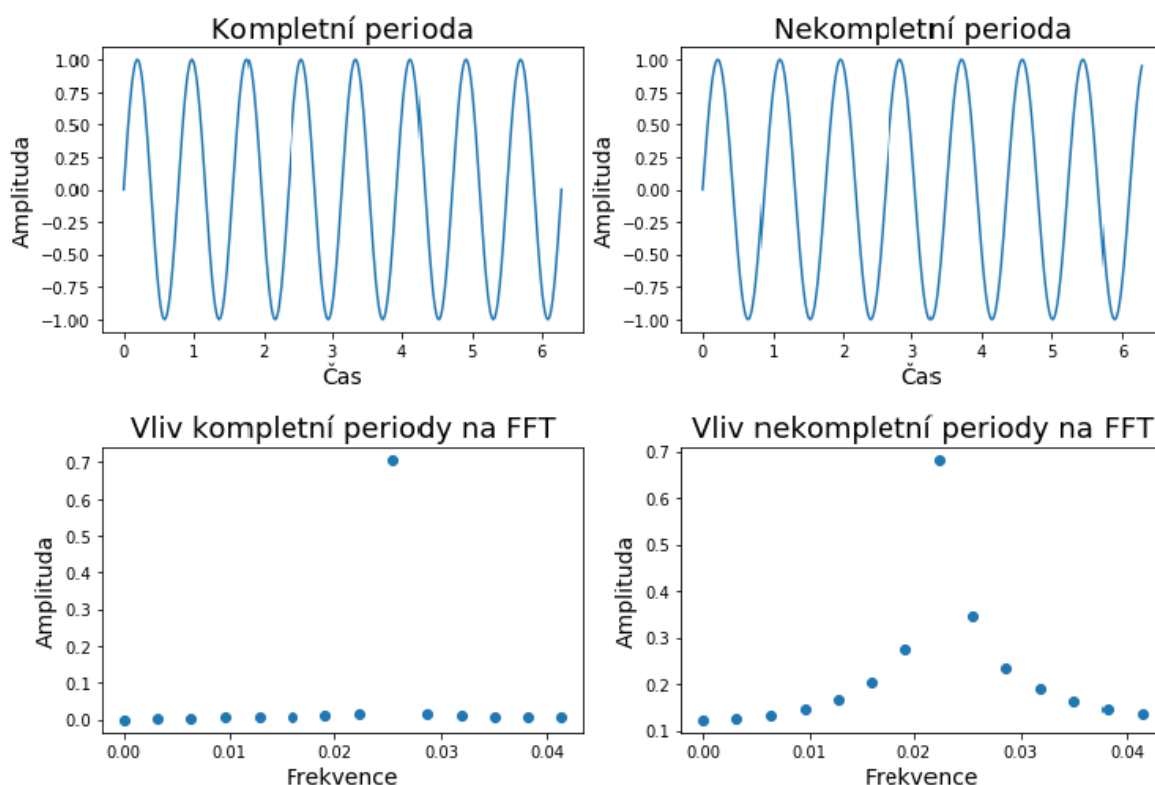
Nevýhodou použitého datového balíčku je, že měření probíhalo pro vícero ložisek současně a během jednoho měření bylo vyhodnoceno vícero poruch, jednotlivé stavy tak nebyly plně shodné, tudíž finální stav pro některá z vyhodnocených ložisek nastal v různém čase a měl vliv na naučená data a rovněž na přesnost vytvořených sítí. Vliv této chyby byl ovšem ve zpracování považován za dostatečně malý.



Obr. 20. Uspořádání testovací sestavy

Byly vytvořeny separátní datasety z měření pro učení a testování výsledků sítě, aby nedocházelo k hodnocení prvku přímo naučeného na síti, a to datasety obsahující pouze OK a NOK data pro vybraná ložiska ze začátku a konce měření a dále záznam celého průběhu života ložiska. Kvůli velikosti záznamu, který měl přes 2000 vzorků měření, bylo vzorkování pro dataset zredukováno na 1 měření každé 0.5, 1 a 2 hodiny, aby bylo možno síť rozumně učit. Data v datasetu samotném nebyla nijak upravována a obsahovala pouze záznam měření a popisek času zaznamenání měření ložiska, pro následné vyhodnocení v programu (*labeled data*).

K úpravě dat docházelo až následně uvnitř programu, aby se předešlo potenciální chybě v různém zpracování uložených formátů. Po importu byla data proložena hammingovým oknem pro omezení vlivu šumu následkem ořezání měřeného signálu (viz. Obr. 21.) a profilována lowpass filtrem.

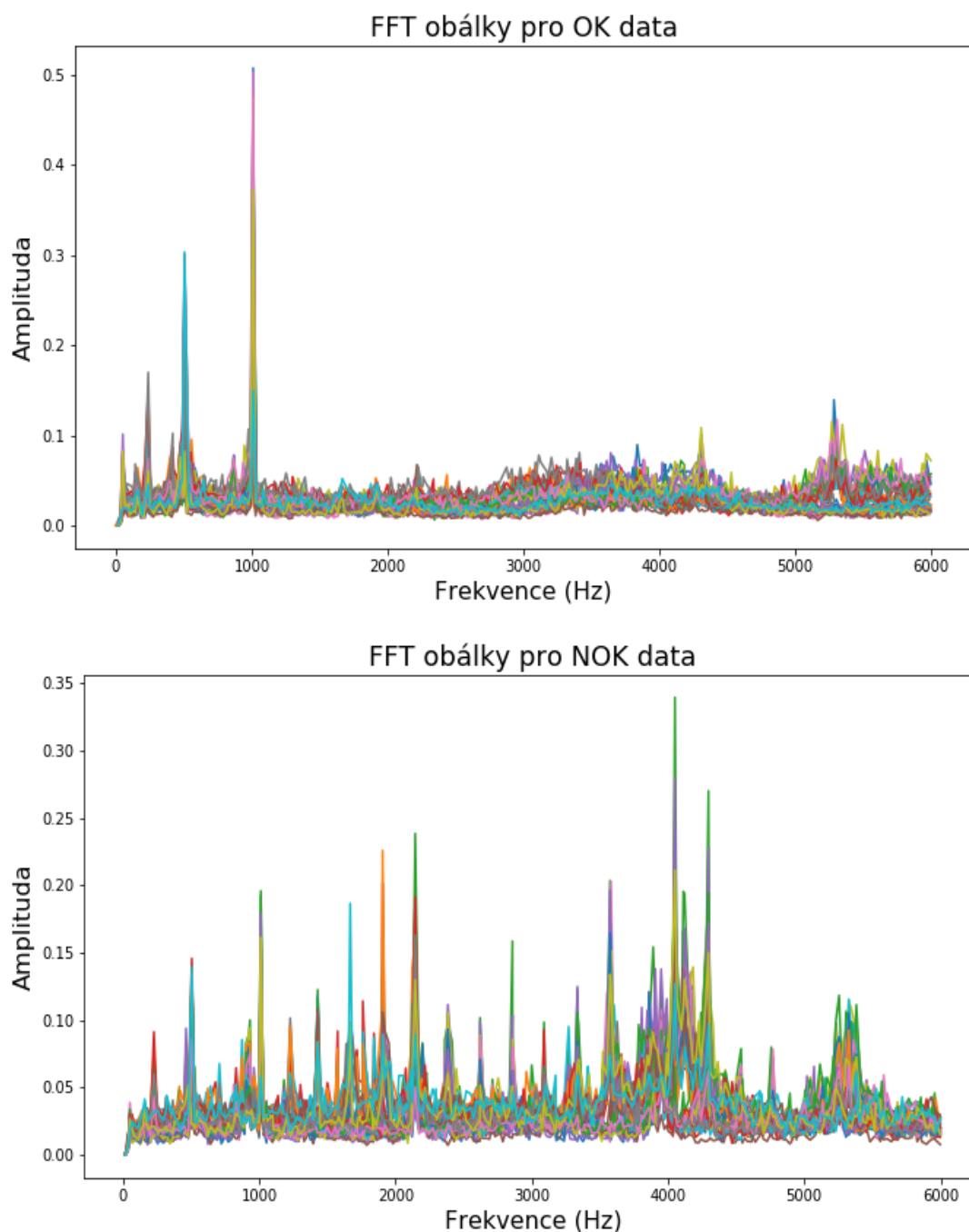


Obr. 21. Vliv ořezané periody na FFT

vlevo patrná pouze jedna špička odpovídající frekvenci, vpravo patrné rozjetí špičky frekvence i do okolí

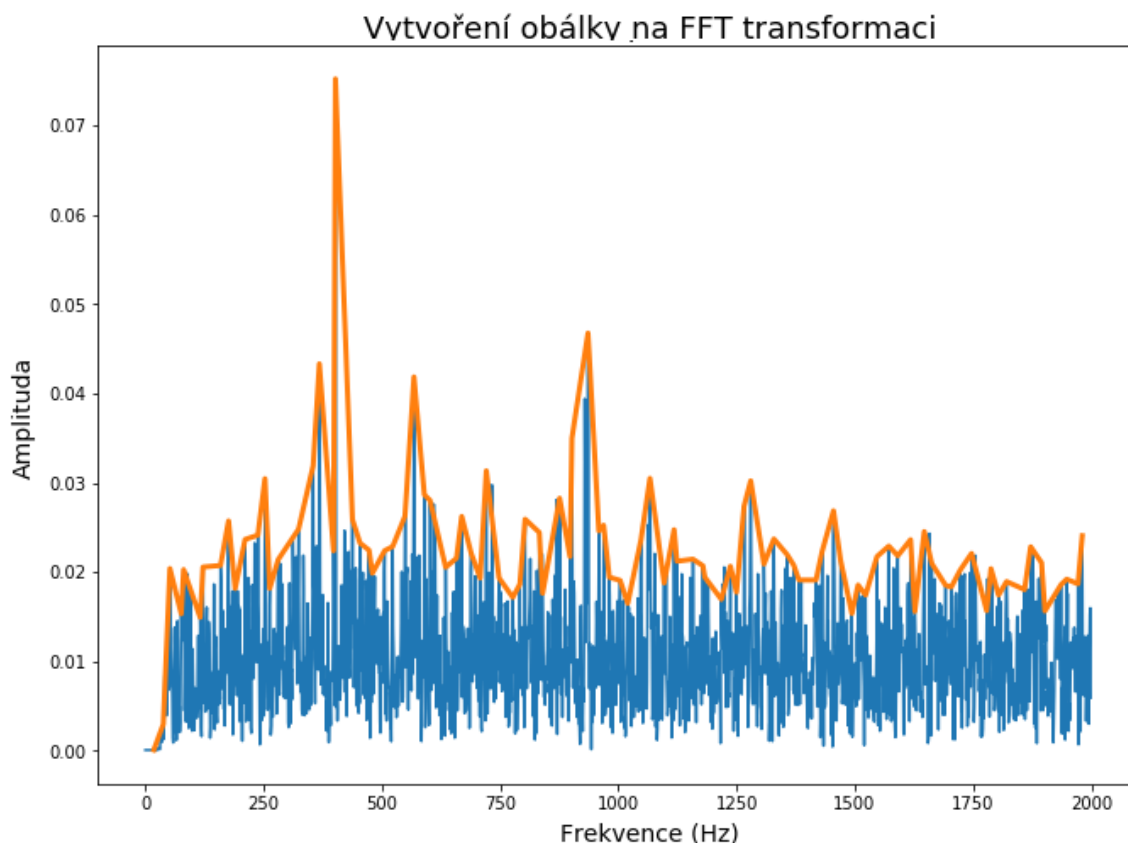
Data pro vyhodnocování byla vybrána ve frekvenci do cca 6000 Hz, kde byl vliv nejvíce pozorovatelný (viz. Obr. 22.), nad touto frekvencí již nebyl vliv degradace ložiska na výstupní signál nijak výrazný.

Jelikož byly analyzovány poruchy na elementech ložiska, byly zajímavé frekvence především nad 10-ti násobkem rotační frekvence hřídele, což odpovídalo frekvencím 330Hz a více. Rovněž byla snaha provádět analýzu pro jednotlivé elementy ložiska a pokusit se určit, jak stav součástí, tak i o poruchu kterého z elementů se primárně jedná. Tato analýza ovšem měla neuspokojivé výsledky, a proto byla zavržena ve prospěch kompletní spektrální analýzy.



Obr. 22. FFT pro OK a NOK data

Jelikož byl vzorek po Fourierově transformaci (viz. Obr. 22.) stále příliš velký co do počtu prvků, pro vložení na vstupní vrstvu sítě, byla provedena redukce pomocí vytvoření obálky, kdy byl zadán požadovaný počet prvků transformace, jakožto velikost vstupu na vstupní vrstvu sítě a byla vytvořena obálka celé transformace (viz. Obr. 23.).



Obr. 23. Výsek z FFT Obálky

Protože bylo, při vytváření obálky, vždy pro dané lokální maximum vybíráno v určitém rozsahu frekvencí, dochází tak ke zkreslení pozice maxima ve frekvenční oblasti, jelikož každé maximum může mít jinou pozici v daném rozsahu, v němž bylo vybíráno (např.: 220–240 Hz). Toto zkreslení však bylo považováno za dostatečně nepatrné v poměru k celému frekvenčnímu rozsahu. Takto vytvořená obálka je následně použita jako vstup do neuronové sítě, kdy vstupní vrstva má šířku rovnající se počtu bodů obálky.

Učení v obou cyklech probíhalo pomocí učitele. Pro oba balíčky dat tedy bylo potřeba vytvořit požadované výstupy pro hodnotící funkci na které by síť byla učená.

5.1.1. Data pro první cyklus učení

Pro první cyklus učení, popsany v následující kapitole, jsou použita označená OK a NOK data (viz. Obr. 22.) měření pro hrubé naučení sítě v první fázi. Data byla vybrána z počáteční a konečné fáze měření. Cílem bylo získat síť schopnou základního ohodnocení předložených vzorků podle jejich přibližného stavu. Pro ověření byl vytvořen menší samostatný dataset s rozdílnými daty od učících dat obsahující rovněž data z počátku a konce záznamu měření na ložisku a síť na něj byla aplikována (viz. Obr. 24.).

OK data pro vyhodnocení stavu součásti. NOK data pro vyhodnocení stavu součásti.

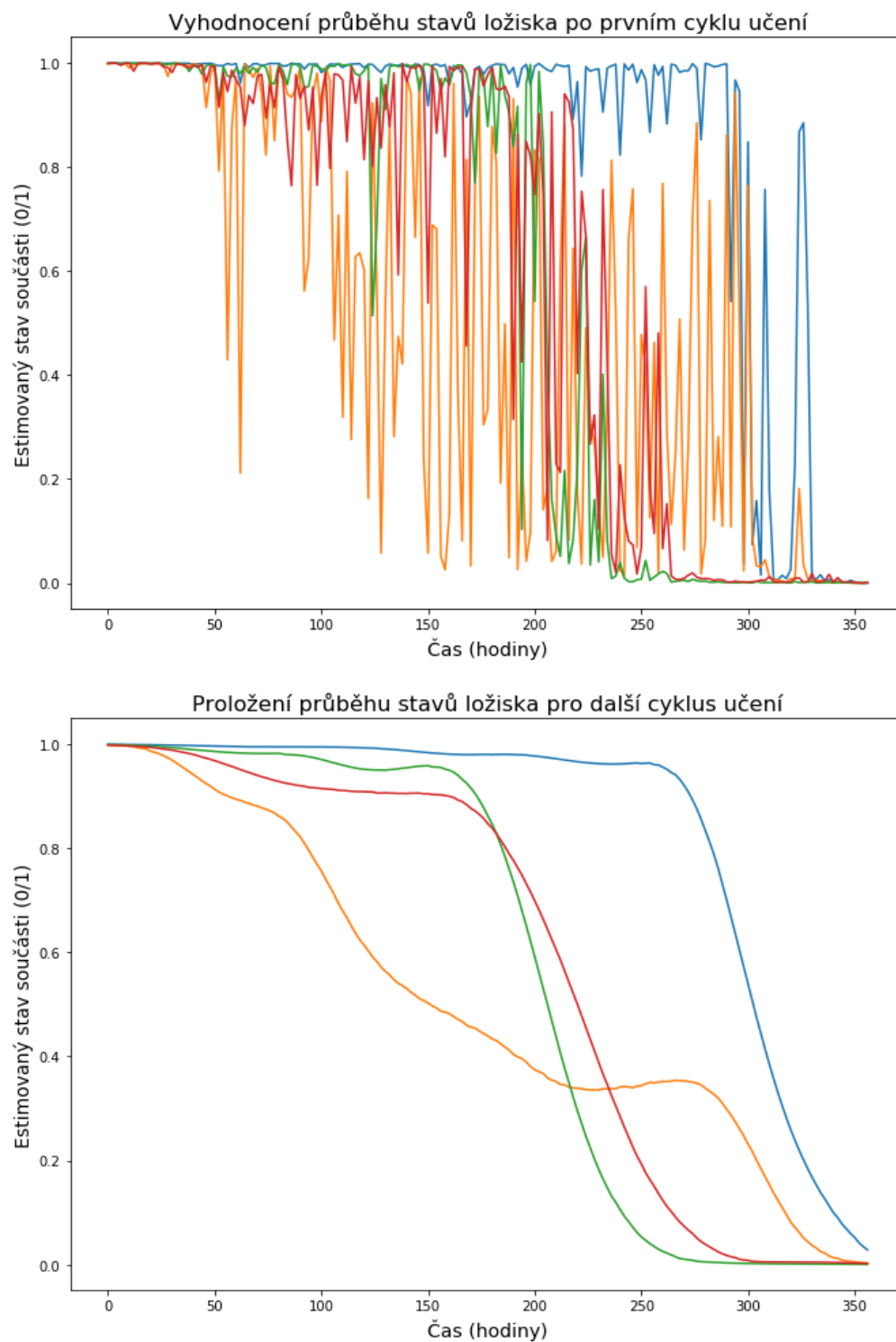
Bearing is OK on	99.7 %	Bearing is KO on	98.0 %
Bearing is OK on	99.2 %	Bearing is KO on	99.0 %
Bearing is OK on	95.9 %	Bearing is KO on	99.9 %
Bearing is OK on	99.9 %	Bearing is KO on	98.8 %
Bearing is OK on	99.7 %	Bearing is KO on	99.4 %
Bearing is OK on	99.7 %	Bearing is KO on	98.7 %
Bearing is OK on	98.9 %	Bearing is KO on	98.8 %
Bearing is OK on	99.2 %	Bearing is KO on	99.7 %
Bearing is OK on	99.6 %	Bearing is KO on	99.7 %
Bearing is OK on	99.8 %	Bearing is KO on	99.1 %
Bearing is OK on	96.1 %	Bearing is KO on	99.7 %
Bearing is OK on	99.5 %	Bearing is KO on	88.8 %
Bearing is OK on	95.3 %	Bearing is KO on	94.3 %
Bearing is OK on	99.5 %	Bearing is KO on	99.9 %
Bearing is OK on	99.4 %	Bearing is KO on	99.7 %

Obr. 24. Vyhodnocení testovacího datasetu OK / NOK dat

Z výsledků je patrné, že síť neměla problém s ohodnocením dat, která pocházela ze začátku a konce průběhu měření, jelikož průběhy obálek byly relativně shodné, což bylo vidno i na vykreslených transformacích (viz. Obr. 22.). Pokud bylo ovšem po síti požadováno vyhodnocení celého průběhu měření, nebyla toho z těchto dat schopna příliš obstojně (viz. Obr. 25.), kde každý bod křivky představuje jeden vyhodnocený záznam, a bylo tedy zapotřebí úpravy dat a druhého cyklu učení na nově označených datech.

5.1.2. Data pro druhý cyklus učení

Pro druhý cyklus jsou použita data z celého průběhu života ložiska. Na ty je aplikována neuronová síť vygenerovaná v prvním kole učícího cyklu a výsledek je profiltrován Hammingovým oknem pro následné použití jako požadovaný výstup pro daná vstupní data. Proložení dat tímto způsobem, je získán hladší průběh křivky popisující průběh stavu ložiska během jeho života. Ačkoli takovéto řešení v první řadě spoléhá na dobře naučenou základní síť, která vytvoří prvořadé hodnocení, lze předpokládat, že degradace ložiska bude probíhat postupně a jeho stav nebude mezi jednotlivými měřeními oscilovat (viz. Obr. 25.). V ideálním případě bude tvar průběhu exponenciální, tento předpoklad ovšem nelze použít jako požadovaný výstup pro učení.



Obr. 25. Nefiltrovaný a filtrovaný průběh degradace ložiska

5.2. Vytvoření neuronové sítě

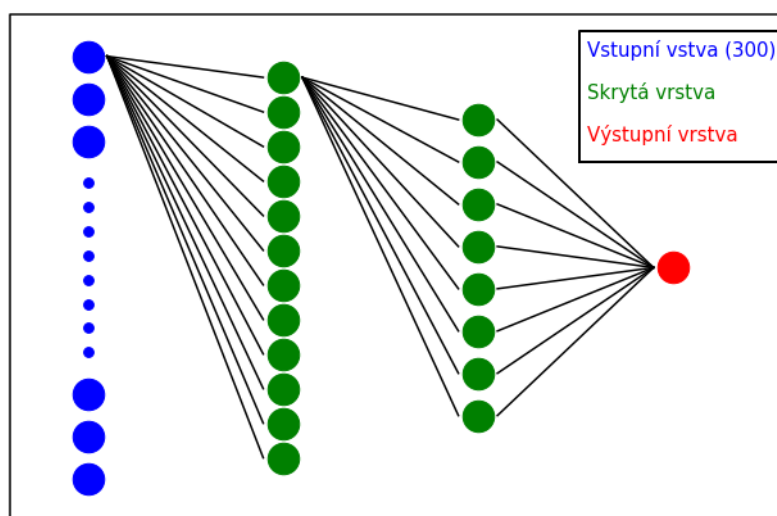
Po importování a zpracování dat je vytvořena neuronová síť, která má vytvořit zjednodušený model k jejich vyhodnocení. Cílem sítě je vytvořit klasifikátor hodnotící stav ložiska na stupnici od 0 do 1, kdy 1 značí funkční ložisko bez poruch a 0 finální poškození ložiska a konec jeho funkčního života. Pro vytvoření a učení sítě nebyla použita žádná z knihoven pro neuronové sítě, pouze základní knihovny s matematickými funkcemi a zpracování signálu. Všechny potřebné funkce pro práci s neuronovou sítí, její tvorbu a učení byly importovány z vlastní napsané knihovny.

5.2.1. Definování sítě

Prvním krokem byla inicializace počátečních podmínek, například velikost kroku učení, počet epoch, nebo velikost učící skupinky (*batch size*), a následně volba samotné sítě.

Při volbě sítě byla možnost importovat a použít již předem vytvořenou síť, která byla již dříve vytvořena a obsahovala uložené rozměry, váhy a aktivační funkce pro jednotlivé vrstvy a tuto síť následně na datech doučit, nebo přeučit na nová data. Druhou možností bylo vytvoření kompletně nové sítě. Možnosti výběru sítě byly nakonec omezeny pouze na základní typ dopředné neuronové sítě, jelikož snahy o použití rekurentních sítí nebyly příliš úspěšné, u níž byla již předdefinována velikost vstupní a výstupní vrstvy, dle importovaného formátu dat a nastavovala se pouze topologie skrytých vrstev (viz. Obr. 26.) a jejich aktivační funkce. Váhy sítě byly následně náhodně vygenerovány symetricky ve zvoleném intervalu (např.: $<0.3, 1>$).

Velikost sítě a učící funkce byly zvoleny experimentálně, buďto ručně, nebo funkcí, jež procházela různá nastavení velikosti sítě a po krátkém učícím cyklu vybírala nastavení s nejmenší dosaženou chybovou funkcí. Tato metoda je ovšem extrémně zdoluhavá a nezaručuje nalezení správné varianty nastavení sítě.



Obr. 26. Jedna z možných topologií sítě
Zde plně propojená s topologií 300-12-8-1

5.2.2. Učení sítě

Učení sítě pro hodnocení stavu ložiska probíhá ve dvou samostatných sekcích, nejprve na označená OK a NOK data a následně na data z průběhu života ložisek (viz. kap. 5.1. *Struktura zpracování problému*). V obou případech se jedná o učení sítě s učitelem (viz. kap. 4.2.1. *Učení s učitelem*). Nastavení sítě je pro obě fáze téměř totožné, až na počáteční podmínky, jelikož se jedná o doučování zvolené sítě. Jednotlivé fáze probíhají v cyklické smyčce, kdy je sledováno možné doučení pod určitou hodnotu chybové funkce, případně po uplynutí maximálního počtu opakování učícího cyklu (*epochs*) a jsou upravovány jednotlivé váhy dle vypočtených přírůstků. V cyklech dochází k následujícím krokům.

Projití vstupních dat pro daný vzorek dopředu sítí. Je vytvořen odhad výstupu na požadovaný vstup. Propagace hodnot přes jednotlivé neurony sítě byla již popsána v předchozí kapitole (viz. kap. 4.1. *Struktura neuronové sítě*).

Z odhadu a požadovaného výstupu je vytvořena chyba, která bude následně zpětně propagována sítí. Určení chyby závisí na zvolené chybové funkci, jelikož každá vrací jiný výsledek (viz. kap. 4.4. *Chybové funkce*). V programu není implementována univerzální chybová funkce, která by byla schopna srovnávat výsledky a nelze tedy výstupy jednotlivých chybových funkcí vzájemně srovnávat. Průběh chyby pro celý vzorek při učení je sledován a je z něj určena hodnotící funkce (*Loss function*) pro sledování průběhu učení v závislosti na počtu uplynulých cyklů.

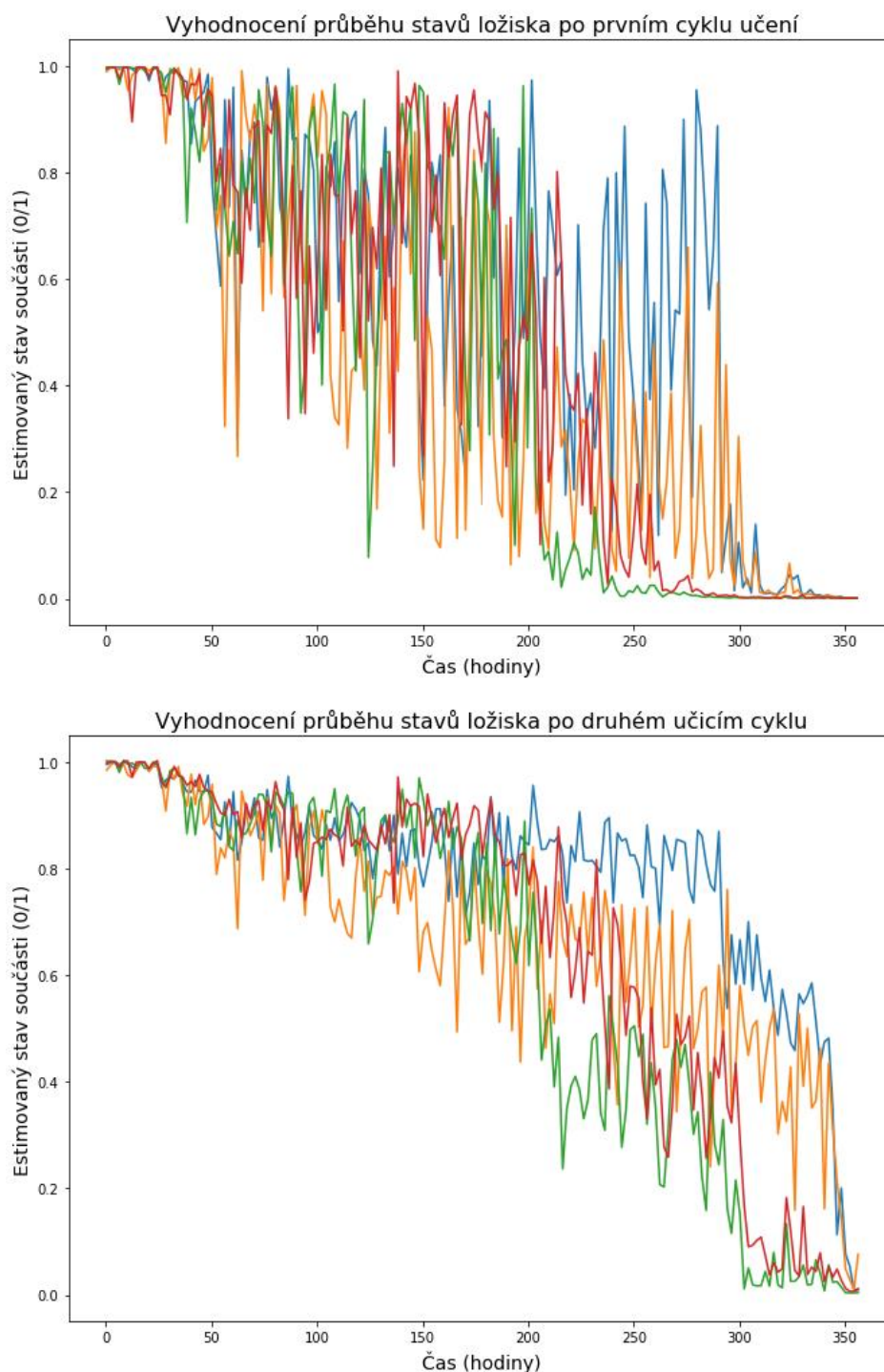
Určená chyba je následně propagována sítí zpět (*Backpropagation*) a jsou určeny jednotlivé přírůstky vah (viz. kap. 4.5. *Backpropagation*). Výstupem je v tomto případě tedy matice jednotlivých přírůstků vah. Jednotlivé váhy mají přiřazeny vlastní setrvačnost pro snadnější překonání případným plytkých lokálních minim, nevýhodou je ovšem zase pomalejší posuv vah v případě prudšího gradientu, kde by se přírůstek samotný pohyboval rychleji a měl tedy kratší výpočetní čas, nebo dokmitávání do minima způsobené setrvačností, jenž má za následek opět prodloužení času doučení. Dále je také regulován pokles rychlosti učení v závislosti na hloubce sítě, kdy váhy blíže vstupní vrstvě se učí pomaleji, než váhy blíže výstupní vrstvě (viz. kap. 4.6. *Learning rate*).

Po projití celé epochy je dále upraven koeficient přírůstku vah (*learn rate*), který má klesající průběh, aby bylo dosaženo počátečního většího přírůstku vah po počátečním náhodném vygenerování a postupný pokles pro snadnější nalezení minima.

Dalšími pomůckami volitelnými při vytváření sítě jsou záměrný postupný pokles vah (*weight decay*), který má za úkol regulovat postupné narůstání vah a odstraňovat spojení, která mají velmi malý přírůstek a jsou tedy zbytečná (viz. kap. 4.6. *Learning rate*), což přímo souvisí s následující funkcí, kterou je odstraňování neuronů s velkým množstvím takto označených vah (viz. kap. 4.8.1. *Prořezávání sítě*) a optimalizovat tak velikost vybrané sítě odstraněním zbytečných neuronů.

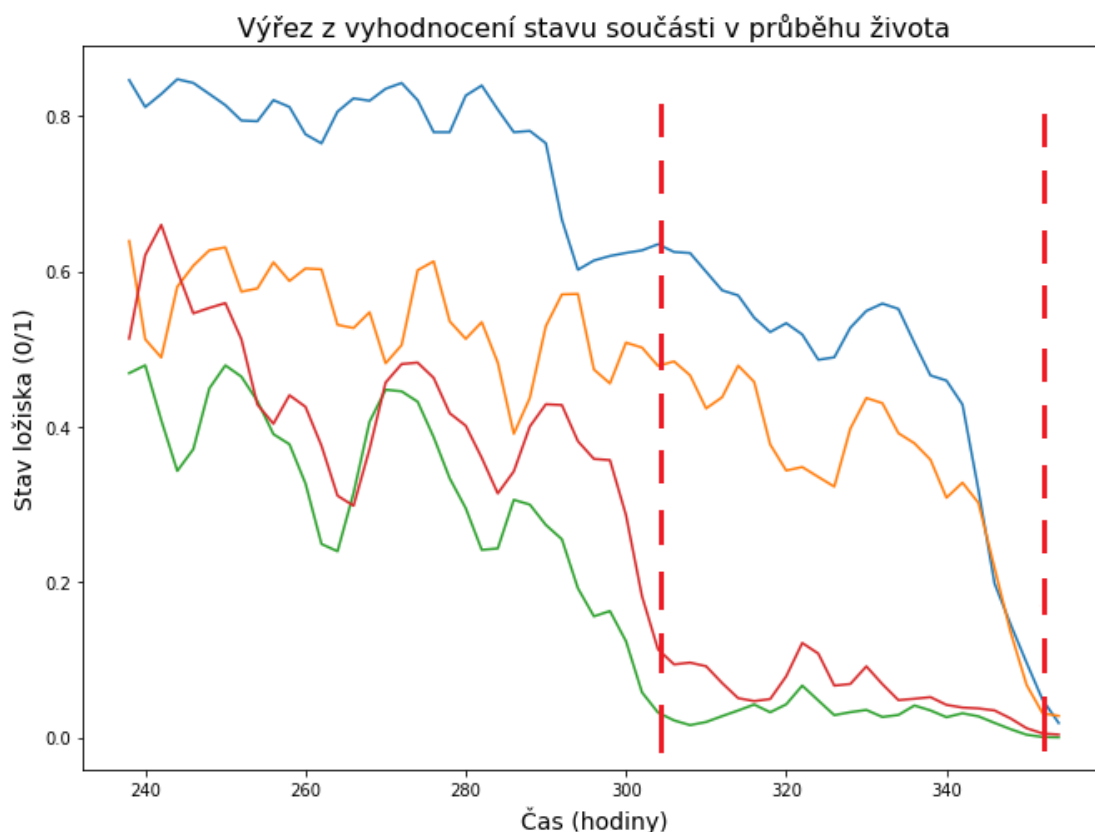
Po naučení je výstup sítě ověřen a testovacích datech, která nejsou shodná s učícími daty, aby výsledek nebyl přímo srovnáván s naučeným vzorkem, čímž by kontrola ztrácela na nezávislosti. Pokud je výsledek uspokojivý, je možno naučenou síť uložit do externího souboru pro pozdější použití a postoupit k dalšímu kolu učení.

Pro druhou fázi učení jsou použita data z průběhu života ložiska, u kterých je snaha zachytit průběh degradace ložiska (viz. kap. 5.2.2. *Data pro druhý cyklus učení*). Cílem této fáze učení je získat síť schopnou, přibližně odhadnout stav ložiska z libovolného vzorku z průběhu života ložiska, jak lze vidět (viz. Obr. 27.), kde je srovnání odhadu sítě před doučením druhým cyklem a po něm. Ačkoli průběh není optimální a bylo by možno se pokusit síť opakovaně doučit k větší přesnosti, lze výstup sítě použít pro následující fázi, již je odhad zbývajících života ložiska (*RUL*).



Obr. 27. Odhad degenerace ložisek
před (nahore) a po (dole) proložení a druhém cyklu učení pro jednotlivá měření

Při doučování sítě v druhé fázi bylo problematické doučit síť na požadovaný průběh hodnot, který by přibližně odpovídal exponenciálnímu průběhu odpovídající průběhu poruchy na ložisku. Primárním problémem bylo vyhodnocení finálního stavu poruchy v měření pro 2 ložiska současně. Při analýze za použití sítě je ovšem patrný rozdíl v proloženém průběhu (viz. Obr. 28.), kdy výskyt finální poruchy je pro jedno ložisko patrný dříve nežli pro druhé, ačkoli byly oběma přisouzeny shodné časy poruchy.



Obr. 28. Průběh života ložiska zobrazující rozdíl ve výskytu finální poruchy vyznačený červenými linkami

Druhým bodem problematiky byla snaha natrénování sítě na co nejmenší rozkmit hodnot průběhu degradace. Toho bylo docíleno pouze částečně (viz. Obr. 27.), kdy hodnoty z průběhu života ložiska mají sice přibližně exponenciální průběh, ale rozkmit je stále relativně velký.

Cílem pro druhou fázi tedy bylo minimalizovat rozkmit hodnot v rámci možností optimalizací sítě, doučením s různými koeficienty, prořezáváním sítě, případně kompletně novým návrhem sítě a jejím naučením. Snahou byla automatizace učení pro 1 a 2 skryté vrstvy a jejich různé kombinace hloubek jednotlivých vrstev.

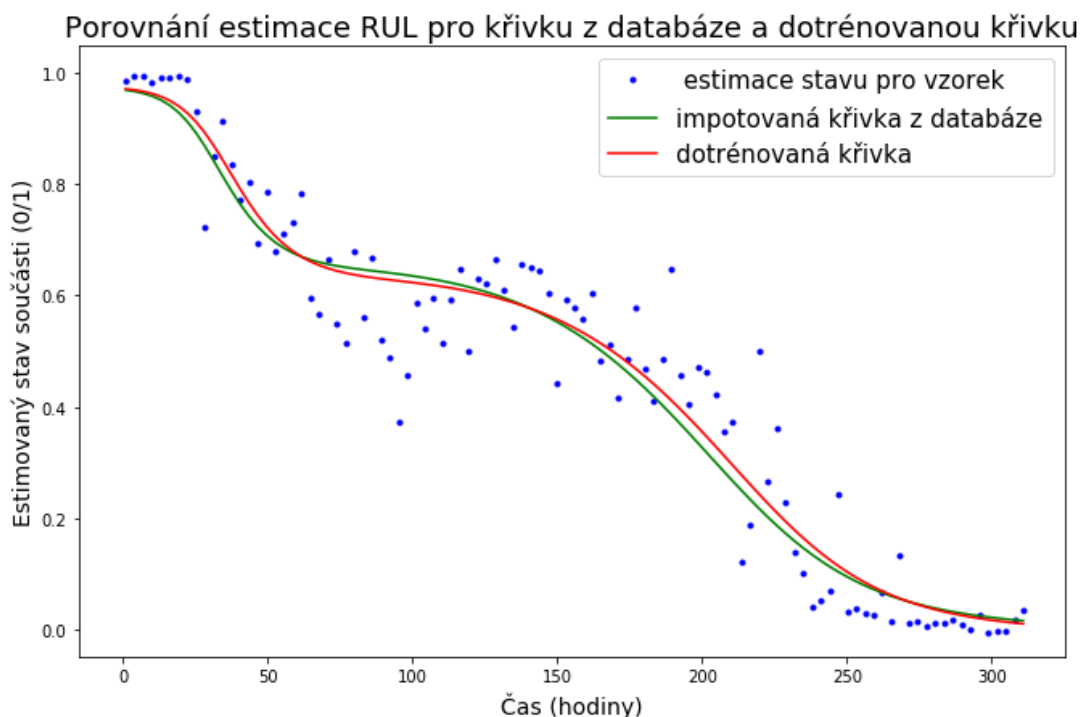
5.3. RUL

RUL (*Remaining Usefull Life*), neboli zbývající užitečný život, je odhad času, po který může stroj fungovat, než bude vyžadována údržba (viz. kap. 4.9. *Odhad zbývajícího života*). Pro vyhodnocení RUL bylo otestováno použití 3 metod, a to následujících.

5.3.1. Run-to-Failure Data

Využívá srovnávání stávajícího průběhu života dané součásti, zde ložiska, s databází průběhů životů obdobných komponent. Z databáze byla následně vybrána křivka, která dosavadnímu průběhu odpovídala nejlépe a na základě této křivky a bodu výskytu její poruchy byl určován RUL. Nevýhodou tohoto řešení je potřeba poměrně rozsáhlé databáze měření pro danou komponentu při obdobném cyklu, aby byla data vzájemně kompatibilní. Pokud by byla metoda s databází aplikována na měření, které nemá společné prvky s předchozími měřeními, nebo je její průběh výrazně odlišný od průběhu křivek v databázi, byl by odhad chybný a nebyl by schopen postihnout možnou blížící se poruchu.

V použitém případě představovala databáze aproximační neuronové sítě, jež byly naučeny na průběh dat z životů jednotlivých měření spolu s hodnotou stavu ložiska při poruše a časem poruchy ložiska. Při porovnávání pak byla vybrána aproximace s nejmenší chybou a po krátkém doučení sítě na dostupná data z dosavadního průběhu stavu ložiska (viz. Obr. 29.) vyhodnocení a odhad zbývajícího času podle místa protnutí doučené křivky s hodnotou výskytu poruchy u předchozího měření.



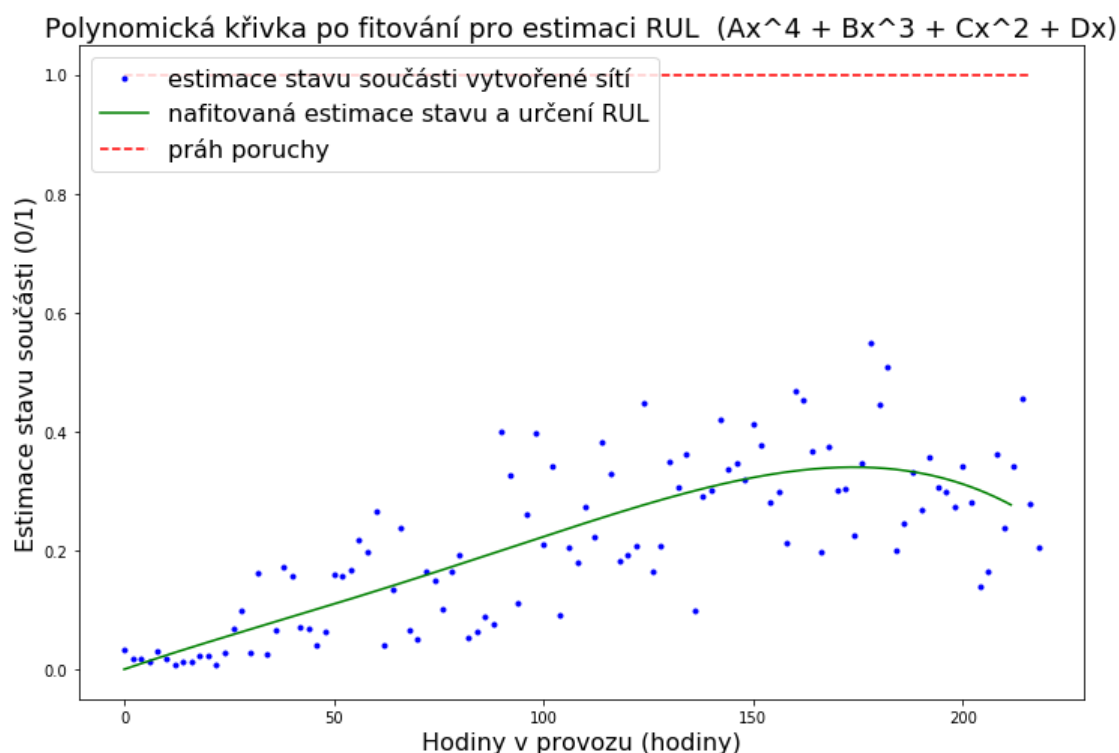
Obr. 29. Importovaná vs doučená křivka pro Run-to-Failure metodu

Nevýhodou této varianty, byla již výše zmiňovaná náročnost na velikost databáze se srovnatelnými daty, bohužel pro tuto diplomovou práci získaný balíček dat byl příliš malý a variabilní v průběžích poruch, stejně jako v délkách časových úseků, mezi poruchami pro měření. Varianta za použití databáze tedy byla možná, a při testování výsledků na nezávislých testovacích datech z měření, která byly ze stejných průběhů životů ložisek, jako křivky v databázích, ale jednotlivé vzorky pro testování nebyly při učení použity, tudíž pro síť představovaly data, která „dříve neviděla“, byly výstupy uspokojivé, ale aplikace by vyžadovala výrazné rozšíření databáze křivek, ze kterých by mohl algoritmus čerpat.

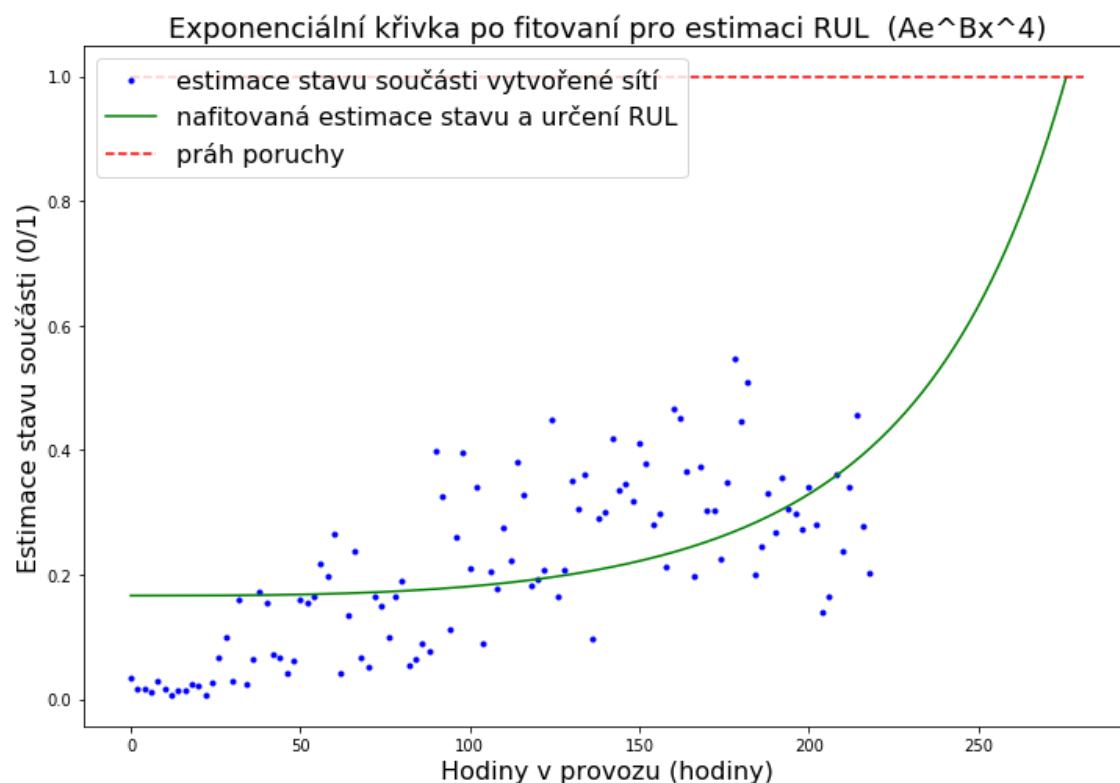
5.3.2. Threshold Data

Metoda využívá předem zvolené hodnoty stavu (*bearing state*), jenž součást nesmí překročit, a aproximací dostupných dat exponenciální křivkou reprezentující průběh degradace se snaží nalézt bod v němž k protnutí dojde a na základě toho určit zbývající čas. Na rozdíl od předchozí metody nevyžaduje rozsáhlou databázi měření z průběhů životů součástí, nebude ovšem schopna poskytnout tak přesný odhad RUL v počáteční fázi jako srovnávání průběhů s předchozími v minulé metodě.

Nutností u metody je nutnost využití exponenciální křivky, na rozdíl od polynomičeských křivek, u kterých není zaručena existence řešení, kterým je protnutí křivky funkce s *threshold* křivkou (viz. Obr. 30.), kde je jasně vidět že po analytickém řešení soustavy rovnic, bylo řešení nevyhovující pro určení RUL. Pro vyhodnocování



Obr. 30. Možná chyba vzniklá při řešení polynomičeské rovnice pro RUL
Hodnota stavu součásti má v tomto grafu obrácenou hodnotu (tedy 0 = OK stav a 1 = NOK stav)



Obr. 31. Řešení pro data s použitím jiné exponenciální křivky

data shodná s daty v Obr. 6.4.2.1.

Hodnota stavu součásti má v tomto grafu obrácenou hodnotu (tedy 0 = OK stav a 1 = NOK stav)

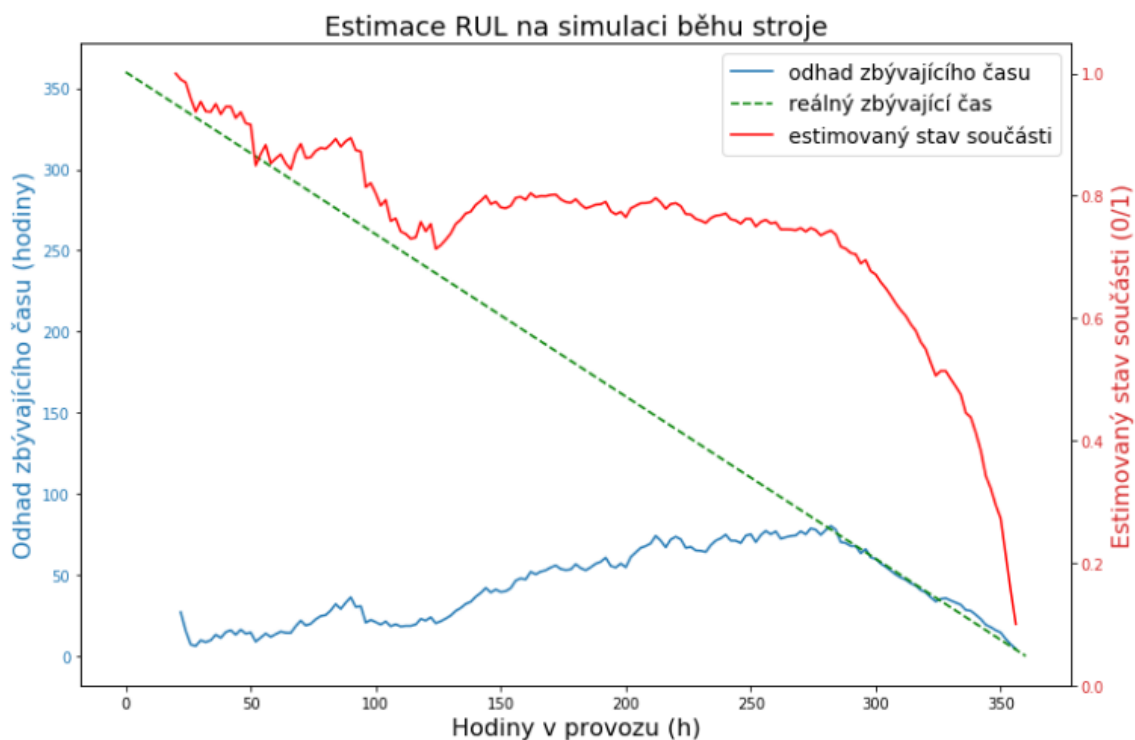
RUL tedy byla využíváno křivek zaručující existenci řešení (viz. Obr. 31.) a případně jejich obdoby. K proložení byl využit model křivky z rovnice (16), kde $f(x)$ je rovnice křivky a jejím výstupem je stav součásti, A, B jsou konstanty určené pomocí estimačního toolboxu a x je proměnná reprezentující čas.

$$f(x) = A \cdot e^{B \cdot x^4} \quad (16)$$

Pro vyřešení fitování křivky bylo využito jak analytického řešení, tak *curve_fit* toolboxu dostupného v základních knihovnách Pythonu. Jelikož získané výsledky se shodovaly, bylo přistoupeno k použití toolboxu, čistě z hlediska nutnosti vytvoření pouze modelu obsahujícího danou křivku pro testování různých variant na rozdíl od ručního řešení a vypsání kódu pro analytické řešení.

Odhad RUL v programu byl proveden pro všechny jednotlivé vzorky měření z průběhu života, simulující tak chování při měření na reálné součásti. Pro prvních 10 vzorků nebylo RUL vyhodnocováno, byl pouze vytvořen odhad stavu součásti, zde ložiska neuronovou sítí vytvořenou a naučenou v předchozím kroku (viz. kap. 5.3. *Neuronová síť*). Pro všechny následující vzorky, vždy pro každý samostatně, již bylo, kromě odhadu stavu, vytvořeno i nafitování křivkou pro určení aktuálního odhadu stavu ložiska a odhadu zbývajících hodin (viz. Obr. 32.).

Ačkoli samotný odhad zbývajících hodin není, v průběhu života, příliš spolehlivý a stabilizuje se více až ke konci života součásti (*modrá křivka v Obr. 32.*), v kombinaci s průběhem odhadu stavu součásti (*červená křivka v Obr. 32.*), jakožto primárního ukazatele stavu součásti, lze relativně obstojně určit přibližující se poruchu. Po každém kroku, neboli přidání měření z průběhu života součásti, je vyhodnoceno, zdali odhadnutý stav součásti nepoklesl pod určitou hodnotu a pokud ano, tak jaký je odhad zbývajících hodin.



Obr. 32. Odhad stavu a RUL součásti (ložiska)

zelená křivka – představuje reálnou hodnotu zbývajícího času

modrá křivka – představuje odhad RUL pro každý přidaný vzorek

červená křivka – představuje odhad aktuálního stavu ložiska pro každý přidaný vzorek

5.3.3. RUL dle skupin

Metoda byla testována jakožto doplňková varianta pro předchozí varianty. Vstupem byl 1D formát měření ve frekvenčním spektru a výstupem byly 3 skupiny s určitým rozsahem zbývajících hodin a to (0-25hodin, 30-55hodin a 60hodin a více). NN síť měla za úkol klasifikaci vstupních dat, do těchto skupin. 1D varianta vstupních dat byla zvolena z důvodu již vytvořeného skriptu zpracovávajícího klasické NN, zatímco pro variantu konvolučních sítí by bylo potřeba vytvořit kompletně novou strukturu, což by bylo velmi náročné a neefektivní vzhledem k požadovanému výstupu z dané části. Výsledek nebyl dostatečně uspokojivý, jelikož síť byla schopna určit data pouze s přibližně 70 %

pravděpodobností (viz. Obr. 33.). Což bylo způsobeno jednak pro tento účel malým trénovacím balíčkem a jednak různým finálním stavem pro různá měření.

Popisek [0h-25h] = 1 [30h-55h] = 2 [60h+] = 3

sk1	sk2	sk3	zbývající čas	odhad skupiny	sk1	sk2	sk3	zbývající čas	odhad skupiny
[0.308 0.337 0.896]			63.0	3	[0.333 0.269 0.9]			66.0	3
[0.251 0.366 0.873]			60.0	3	[0.207 0.342 0.894]			63.0	3
[0.234 0.321 0.848]			57.0	3	[0.117 0.514 0.832]			60.0	3
[0.229 0.45 0.749]			54.0	3	[0.145 0.411 0.831]			57.0	3
[0.169 0.547 0.69]			51.0	3	[0.08 0.73 0.41]			54.0	2
[0.234 0.54 0.703]			48.0	3	[0.104 0.624 0.76]			51.0	3
[0.35 0.575 0.528]			45.0	2	[0.068 0.83 0.336]			48.0	2
[0.177 0.552 0.747]			42.0	3	[0.018 0.937 0.316]			45.0	2
[0.19 0.74 0.492]			39.0	2	[0.017 0.97 0.255]			42.0	2
[0.117 0.877 0.332]			36.0	2	[0.046 0.946 0.156]			39.0	2
[0.18 0.647 0.626]			33.0	2	[0.108 0.838 0.31]			36.0	2
[0.515 0.669 0.157]			30.0	2	[0.405 0.728 0.077]			33.0	2
[0.45 0.456 0.507]			27.0	3	[0.534 0.441 0.447]			30.0	1
[0.687 0.321 0.331]			24.0	1	[0.797 0.149 0.298]			27.0	1
[0.65 0.287 0.398]			21.0	1	[0.8 0.243 0.18]			24.0	1
[0.543 0.503 0.353]			18.0	1	[0.77 0.526 0.053]			21.0	1
[0.718 0.211 0.465]			15.0	1	[0.857 0.242 0.097]			18.0	1
[0.791 0.29 0.316]			12.0	1	[0.872 0.314 0.06]			15.0	1
[0.969 0.056 0.177]			9.0	1	[0.995 0.034 0.015]			12.0	1

Obr. 33. Výstup pro síť groups

červeně označeny chybné odhady (odhad výstupů pro poslední skupinu 0-25h byl uspokojivý)

Pro učení byla využita diskriminační matice (viz. Obr. 34.), která byla aplikována na vyhodnocovanou chybu za účelem posílení vlivu kritických stavů na úkol stavů, ve kterých byl dostatek času. Důvodem byla potřeba predikovat s větší jistotou stavy blízké poruše, jelikož předčasná výměna dílu, má menší negativní následky než předčasný stav poruchy, jenž by měl za následek neplánovanou odstávku.

		Predikce			Skutečná hodnota
		3	2	1	
skupiny	3	0.75	0.2	0.05	
	2	0.1	0.35	0.55	
	1	0.05	0.1	0.85	

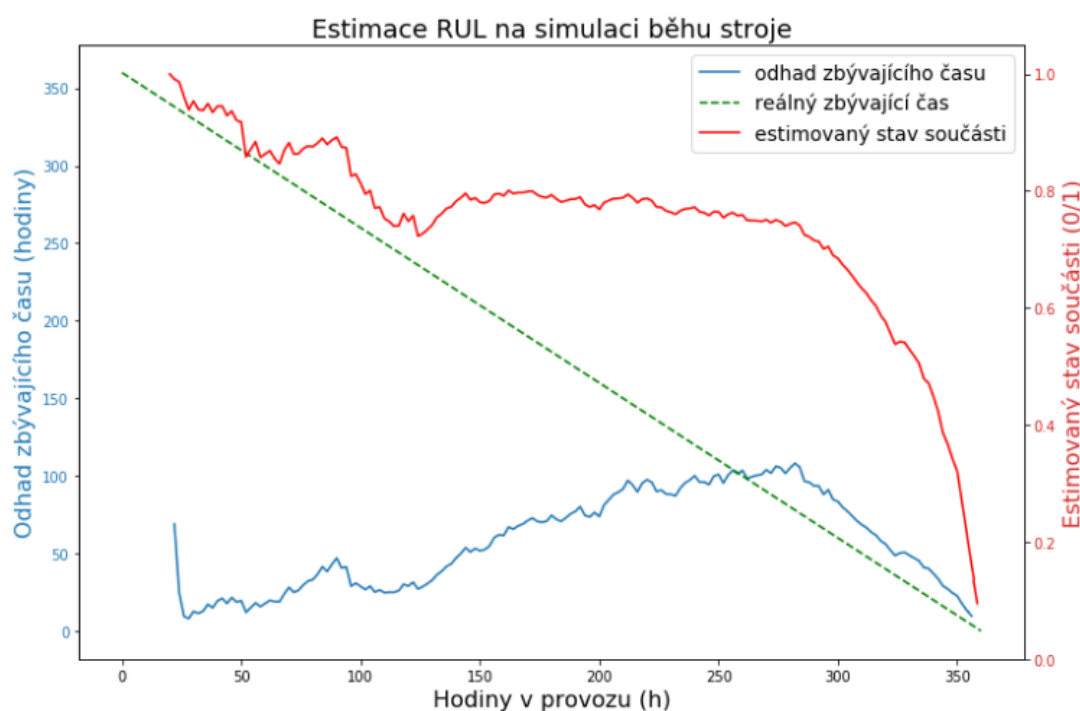
Obr. 34. Diskriminační matice

[0h-25h] = 1.skupina [30h-55h] = 2.skupina [60h+] = 3.skupina

5.3.4. Vyhodnocení RUL

Metoda Run-to-Failure byla zavržena z důvodu velmi malé databáze měření průběhu života ložiska. Její využitelnost by mohla nastat s postupným narůstajícím objemem provedených měření. Metoda RUL dle skupin byla zvažována jakožto záložní identifikátor v případě blížící se poruchy, ovšem její funkčnost rovněž nebyla uspokojivá z důvodu udaného konce života ložisek v různých fázích, což mělo za následek, že se podoba finálních dat lišila a síť nebyla schopna ji dostatečně postihnout.

Pro vyhodnocení tedy byla zvolena threshold metoda. Vstup dat byl simulován pro měření přicházející z monitorovaného stroje. Programu tak byly přikládány jednotlivé vzorky samostatně k vyhodnocení a výsledek byl postupně vykreslován (viz. Obr. 35. a Obr. 36.). Primárním indikátorem pro vyhodnocení RUL byl odhadnutý stav součásti pro každé měření (červená křivka v Obr. 35. a Obr. 36.) a jeho následná hodnota po proložení exponenciální křivkou a hodnota protnutí exponenciální křivky s prahovou hodnotou poruchy (viz. Obr. 30. a Obr. 31.) a z podílu místa protnutí a hodnoty času posledního měření je určena hodnota RUL (modrá křivka v Obr. 35. a Obr. 36.).



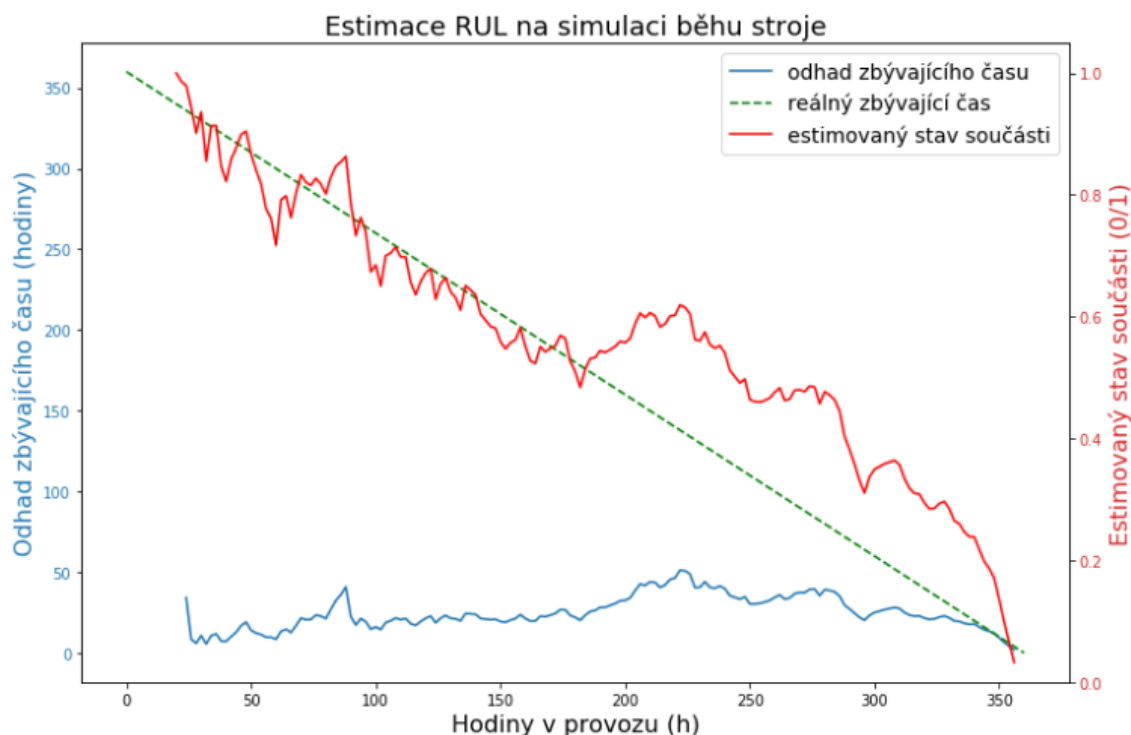
Obr. 35. Odhad stavu a RUL součásti (ložisko 1)

zelená křivka – představuje reálnou hodnotu zbývajících času

modrá křivka – představuje odhad RUL pro každý přidávaný vzorek

červená křivka – představuje odhad aktuálního stavu ložiska pro každý přidávaný vzorek

Z grafů je patrné, že odhad RUL je po většinu času nespolehlivý, kdy u ložiska 1 začne odpovídat realitě až nějakých cca 80 hodin před poruchou a u ložiska 2 pouze nějakých 30 hodin před poruchou. U obou variant je ovšem hodnota stavu součásti a její průběh relativně uspokojivým ukazatelem stavu. V kombinaci se samotným odhadem RUL tedy síť poskytuje varování před poruchou.



Obr. 36. Odhad stavu a RUL součásti (ložisko 2)

zelená křivka – představuje reálnou hodnotu zbývajícího času

modrá křivka – představuje odhad RUL pro každý přidáný vzorek

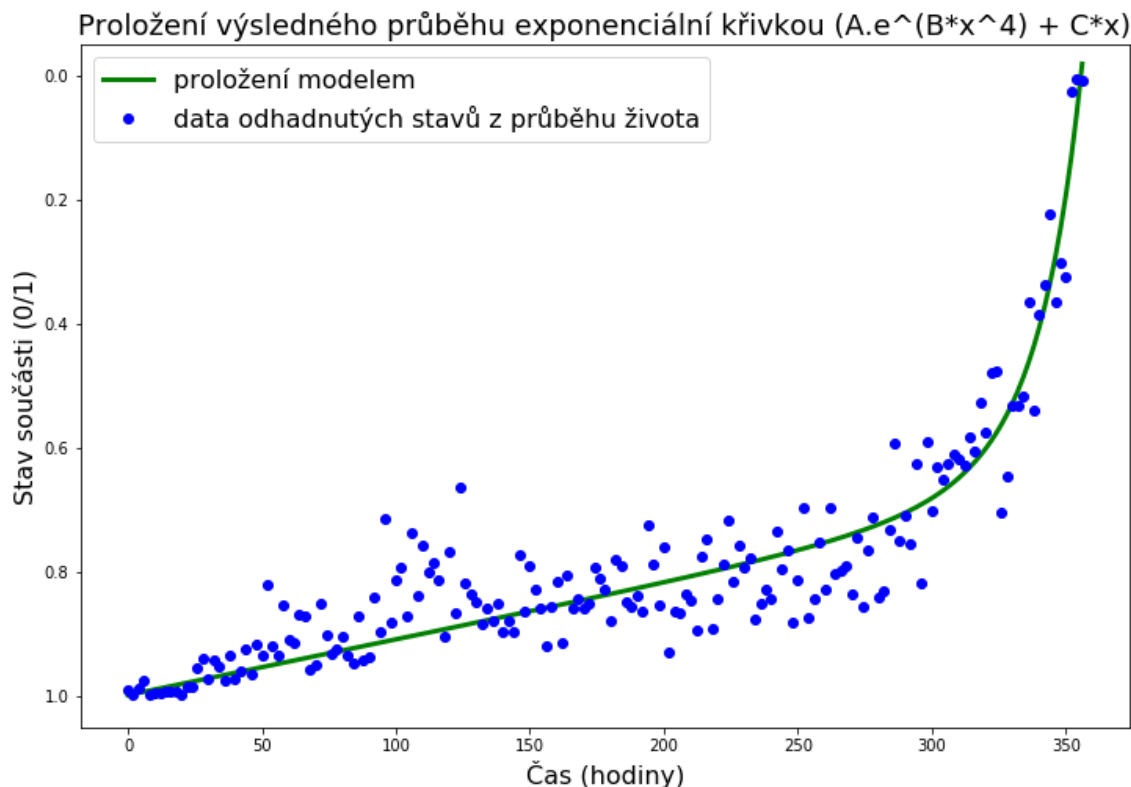
červená křivka – představuje odhad aktuálního stavu ložiska pro každý přidáný vzorek

5.3.5. Doučování sítě

Jednou z výhod neuronových sítí, oproti lineárním modelům, je možnost doučování a zpřesňování sítě za chodu s narůstajícím objemem zpracovaných dat při funkci na daném stroji.

Řešení doučování bylo v práci pojato jako učení s učitelem po ukončení vyhodnocování stavu součásti pro jeho daný cyklus života. Doučování tedy bylo prováděno až po ukončení monitorování dané součásti, aby nedocházelo k ovlivňování výsledků v samotném průběhu monitorovacího cyklu. Data byla po ukončení proložena exponenciálním modelem (viz. Obr. 37.) a proložení bylo využito jako učitel při doučování modelu samotného.

Učení bylo prováděno s menším koeficientem učení a na menším počtu cyklu, aby nedošlo k přílišnému ovlivnění výstupu sítě z jednoho vyhodnocení výsledku. Topologie modelu rovněž nebyla již nijak upravována při doučování. Při doučování ovšem vyvstávala obava, že při nesprávném automatickém proložení dat křivkou, která je následně používána jako učitel může výkonnost sítě spíše klesat nežli stoupat. Váhy se tedy při doučování dají ukládat extra, aby bylo možno zpětně použít nepozměněnou síť.



Obr. 37. Proložení odhadů stavů součásti pro doučování modelu

K proložení byl využit model exponenciální křivky, z rovnice (17), kde $f(x)$ je rovnice křivky a jejím výstupem je stav součásti, A, B, C jsou konstanty určené pomocí estimačního toolboxu a x je proměnná reprezentující čas.

$$f(x) = A \cdot e^{B \cdot x^4} + C \cdot x \quad (17)$$

Pro proložení nebyla využita shodná křivka s křivkou pro estimaci RUL, jelikož daná exponenciální křivka nemá počátek v 0, kudy je ovšem potřeba, aby počáteční bod procházel, jelikož se jedná o výchozí stav součásti. Křivka ovšem nemá lineární řešení a proměnnou x tedy nelze analyticky vyjádřit, aby bylo možno v průběhu měření nalézt řešení pro jednotlivé hodnoty proložení, proto bylo od jejího použití upuštěno ve prospěch křivky dle rovnice (16).

6. Zhodnocení výsledků a diskuze

Součástí práce bylo otestování vytvořené neuronové sítě na reálných datech. Vzhledem k nastalé situaci byla zvolena data z vibračních měření na ložiscích z veřejně přístupné databáze. Měření obsahovalo záznam vibrací z průběhu celého života ložiska. Cílem vytvořené neuronové sítě bylo postupně pro každé měření určit stav součásti, ze kterého byl následně odhadnut zbývající život. Informaci o stavu dané součásti, a její zbývající čas do poruchy by bylo následně možno využít při plánování prediktivní údržby stroje s ohledem na maximální využití života součásti a minimalizaci prostoje pro potřebnou výměnu.

Pro vyhodnocování dat byla zvolena dopředná neuronová síť s architekturou MLP v topologii 300-80-1 s aktivačními funkcemi sigmoid, jejíž výstupem byl odhad stavu součásti. Tato architektura byla zvolena z důvodu jejího obvyklého použití pro vyhodnocování daného typu dat. Topologie byla vybrána experimentálně pomocí algoritmu testujícího různé varianty topologií a aktivačních funkcí a srovnávajícího dosažených výsledků.

Výsledkem byla knihovna, pomocí níž byla vytvořena neuronová síť schopná odhadu stavu součásti a algoritmus predikce zbývajícího života součásti (viz. *Obr. 32. Obr. 35 a Obr. 36.*). Dosažené výsledky odhadu zbývajícího času života byly podmíněny dostupnými daty, jejichž objem byl malý, což kompromitovalo schopnost sítě generalizovat data na úkor variací v jednotlivých záznamech. Jelikož dostupná data byla pouze záznamem z průběhu života součásti, bez indikátoru reálného stavu na základě kterého, by bylo možno síť učit. Bylo tedy využito postupů pro učení sítě popsaných v předchozích kapitolách.

Námětem pro další práci by mohlo být doplnění knihovny o další varianty architektur a jejich srovnání s použitou variantou. Vytvoření obsáhlejšího datasetu, na němž by bylo možno vytvářené a testované sítě lépe učit.

7. Závěr

Cílem diplomové práce bylo seznámit se s tématem umělých neuronových sítí a následně vytvořit knihovnu pro jejich tvorbu a její využití pro vytvoření neuronové sítě. Daná neuronová síť bude následně využita k monitorování technického stavu stroje vyhodnocováním měřených dat. Z technického stavu bude dále určen zbývajících život vyhodnocované součásti pro plánování prediktivní údržby.

V úvodní části práce je představena struktura rozebírané problematiky a význam použití neuronových sítí.

V dalších částech je rozebrána struktura neuronových sítí, spolu s jednotlivými prvky proměnných volenými při učení neuronové sítě a jejich implementace do vytvořené knihovny. Dále je popsáno vypracování sítě pomocí této knihovny, zpracování dat pro její učení a optimalizace topologie. Na základě vytvořené sítě je následně implementován algoritmus odhadu zbývajících života součásti pro prediktivní údržbu.

Knihovna, neuronová síť i algoritmus odhadu zbývajících života byly napsány v jazyce Python pomocí editoru Jupyter za využití základních matematických knihoven. Software je tedy rozdělen na knihovnu obsahující funkce potřebné pro vytvoření a učení neuronové sítě a na samotné algoritmy, z nichž je knihovna volána a jsou jí pouze předkládány požadované proměnné.

Jedním z problémů byla volba vhodných dat a jejich úprava. Jelikož na datech z dostupných měření byly současně měřeny vibrace na vícero ložiscích, docházelo při snímání k vzájemnému ovlivňování, a rovněž finální stav byl na datech z průběhu života součástí vyhodnocen pro více ložisek současně, což mělo za následek ze teoretické stavy poruch nebyly pro jednotlivá ložiska shodné.

Cíle práce byly splněny v plném rozsahu, a to v následujících bodech.

Bylo provedeno seznámení se s neuronovými sítěmi a jejich interní funkcionalitou.

Byla napsána knihovna pro vytvoření, učení a testování umělé neuronové sítě a algoritmus odhadu zbývajících života monitorované součásti.

Tato knihovna byla následně využita pro navržení, učení a optimalizaci topologie umělé neuronové sítě pro monitorování stavu součásti stroje a odhad jeho zbývajících života.

Navržená knihovna byla ověřena na datech z reálného měření na součásti, zde ložisku, z volně přístupné databáze.

8. Seznam zdrojů

8.1. Použitá literatura

- [1] Maintenance Strategies to Reduce Downtime Due to Machine Errors. *ResearchGate* [online]. c2014. [cit. 2020-06-16]. Dostupné z: https://www.researchgate.net/publication/267864966_Maintenance_Strategies_to_Reduce_Downtime_Due_to_Machine_Positional_Errors
- [2] PELLICCIONE, Amanda. PREDICTIVE MAINTENANCE STUDY: Process industry PdM trends. *Plant Engineering* [online]. c2018. [cit. 2020-06-04]. Dostupné z: <http://search.proquest.com/docview/2164476792/>
- [3] A Beginner's guide to Neural Networks and Deep Learning. *A.I.Wiki* [online]. c2019. [cit. 2020-02-17]. Dostupné z: <https://pathmind.com/wiki/neural-network>
- [4] Supervised vs Unsupervised Learning, *towards data science* [online]. c22.5.2018 [cit. 2020-02-25] Dostupné z: <https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d>
- [5] Březina, Tomáš CSc. *Aktivační dynamiky*. Brno, 2019. Dostupné z: intranetu VUT FSI. Interní skripta
- [6] 5 Regression Loss Functions All Machine Learners Should Know. *Heart Beat* [online]. c5.6.2018 [cit. 2020-02-23]. Dostupné z: <https://heartbeat.fritz.ai/5-regression-loss-functions-all-machine-learners-should-know-4fb140e9d4b0>
- [7] Quantile loss function for machine learning. In: *TECH BLOG* [online]. c2016-2020 [cit. 2020-02-23]. Dostupné z: <https://www.evergreeninnovations.co/blog-quantile-loss-function-for-machine-learning/>
- [8] 3Blue1Brown. Backpropagation calculus | Deep learning, chapter 4. In: *Youtube* [online]. c3.11.2017 [cit. 2020-02-17]. Dostupné z: <https://youtu.be/tleHLnjs5U8>
- [9] How to Configure the Learning Rate When Training Deep Learning Neural Networks. *Machine Learning Mastery* [online]. c23.1.2019 [cit. 2020-02-23]. Dostupné z: <https://machinelearningmastery.com/learning-rate-for-deep-learning-neural-networks/>
- [10] Vibration Analysis Explained. *ReliablePlant* [online]. c2020. [cit. 2020-06-08]. Dostupné z: <https://www.reliableplant.com/vibration-analysis-31569#:~:text=Vibration%20analysis%20is%20defined%20as,machines%20and%20their%20components%20are.&text=Vibration%20analysis%20can%20detect%20problems,Bearing%20failures>

- [11] SKF GROUP. SKF: *Super-precision bearings*. Gothenburg, Sweden, c2016. [cit. 2020-06-18]. Dostupné z: http://www.skf.com/binary/21-129877/Superprecision-bearings-catalogue---13383_2-EN.pdf
- [12] COLAH. Neural Networks, Manifolds, and Topology. In: *Colah's blog* [online]. c6.5.2014 [cit. 2020-02-23]. Dostupné z: <https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/>
- [13] How to Control the Stability of Training Neural Networks With the Batch Size. *Machine Learning Mastery* [online]. c10.1.2020 [cit. 2020-02-23]. Dostupné z: <https://machinelearningmastery.com/how-to-control-the-speed-and-stability-of-training-neural-networks-with-gradient-descent-batch-size/>
- [14] Mehlig B. *Artificial Neural Networks*. Goteborg, 2019. c1.2.2019 [cit. 2020-02.23]. University of Gothenburg, Department of Physics.
- [15] Three ways to Estimate Remaining Useful Life for Predictive Maintenance. *Mathworks* [online]. c2018. [cit. 2020-06-08]. Dostupné z: <https://www.mathworks.com/company/newsletters/articles/three-ways-to-estimate-remaining-useful-life-for-predictive-maintenance.html>
- [16] Remaining Life Estimation with Keras. *TowardsDataScience* [online]. c2019. [cit. 2020-06-08]. Dostupné z: <https://towardsdatascience.com/remaining-life-estimation-with-keras-2334514f9c61>
- [17] J.Lee, H.Qiu, G.Yu, J.Lin, and Rexnord Technical Services. University of Cincinnati. „Bearing Data Set“. *NASA Ames Prognostics Data Repository*. [online]. c2007. [cit. 2020-06-08]. Dostupné z: <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository>
- [18] SEMOTAM, P. Prediktivní systém údržby obráběcích strojů s využitím vibrodiagnostiky. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2018. 109 s. Vedoucí diplomové práce doc. Ing. Miloš Hammer, CSc.
- [19] Industrial application of neural networks – an investigation. *ScienceDirect* [online]. c2001. [cit. 2020-06-07]. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0959152400000275>
- [20] Data4Bio. Data Fitting: Basic Curve Fitting, part 2. In: Youtube [online]. c10.6.2016 [cit. 2020-06-08]. Dostupné z: <https://youtu.be/5bCM5hcr5k4>
- [21] Wind Turbine High-Speed Bearing Prognosis. *MathWorks* [online]. c2020. [cit. 2020-06-08]. Dostupné z: <https://www.mathworks.com/help/predmaint/ug/wind-turbine-high-speed-bearing-prognosis.html>

- [22] *Industrial application of neural network*. s. 506. In: *ScienceDirect* [online]. c2001 [cit. 2020-06-03]. Dostupné z: <https://www.sciencedirect.com/science/article/pii/S0959152400000275>
- [23] Bishop, Christopher M. *Pattern Recognition and Machine Learning*. s. 758. Springer Media. c2006. [cit. 2020-06-18]. ISBN-13: 978-0387-31073-2
- [24] Artificial Neural Network based Condition Monitoring of Rolling Element Bearing using Vibration Signals. *ResearchGate* [online]. c2011 [cit. 2020-06-08]. Dostupné z: https://www.researchgate.net/publication/263172110_Artificial_Neural_Network_based_Condition_Monitoring_of_Rolling_Element_Bearing_using_Vibration_Signals

8.2. Obrázky

- [1.] Industry 4.0. In: *Wikipedia* [online]. c2019. [cit. 2020-06-08]. Dostupné z: https://upload.wikimedia.org/wikipedia/commons/7/7a/Industry_4.0_%28cs%29.png
- [10.] Data Overfitting. In: *Wiki* [online]. 02.2008 [cit. 2020-02-23]. Dostupné z: <https://upload.wikimedia.org/wikipedia/commons/1/19/Overfitting.svg>
- [11.] Angular contact ball bearing. In: *LandEfeld*. [online]. c2019. [cit. 2020-06-08]. Dostupné z: https://www.landefeld.com/shop/media/technik/z_schraegkugellager_einreihig.jpg
- [12.] Compression and Acceleration of High-dimensional Neural Networks, In: *Intel* [online]. c2018. [cit. 2020-06-16]. Dostupné z: <https://software.intel.com/content/dam/develop/external/us/en/images/webops11243-fig4-pruning-799917.png>
- [14.] Aproximování na databázi. In: *MathWorks* [online] 2018 [cit. 2020-05-31]. Dostupné z: https://www.mathworks.com/content/mathworks/www/en/company/newsletters/articles/three-ways-to-estimate-remaining-useful-life-for-predictive-maintenance/_jcr_content/mainParsys/image_0_copy_copy_co_1738644073.adapt.full.high.jpg/1566909946997.jpg
- [15.] Threshold data. In: *MathWorks* [online] 2018 [cit. 2020-05-31]. Dostupné z: https://www.mathworks.com/content/mathworks/www/en/company/newsletters/articles/three-ways-to-estimate-remaining-useful-life-for-predictive-maintenance/_jcr_content/mainParsys/image_0_copy_copy_co_1127560020.adapt.full.high.jpg/1566909947025.jpg
- [20.] Uspořádání testovací sestavy. In: *ResearchGate* [online] 09.2005 [cit. 2020-05-31]. Dostupné z: https://www.researchgate.net/profile/Kwangsuk_Lee/publication/232132253/figure/fig3/AS:699706476875779@1543834432720/The-test-rig-of-IMS-bearing-dataset.jpg

9. Seznam zkratk, symbolů a obrázků

9.1. Seznam symbolů

m [Kg] – hmotnost

[N/m] – tuhost

b [Ns/m] – tlumení

Q [-] – buzení

ζ [-] – tlumení

ω_n [rad/s] – úhlová frekvence

f_n [Hz] - frekvence

rpm [ot/min] – rychlost otáčení

n [-] – počet valivých elementů

B_d [mm] – průměr valivého ložiska

P_d [mm] – roztečný průměr ložiska

Φ [deg] – kontaktní úhel

$x_I - x_n$ [-] – hodnoty z předchozí vrstvy vstupující do neuronu

$w_I - w_n$ [-] – váhy na vstupu neuronu

y [-] – výstupní hodnota ze sítě

y_p [-] – požadovaná výstupní hodnota

δ [-] – parametr zlomu křivky Pseudo-Huber

q [-] – parametr poměru rozdělení dat pro Quantile

Co [-] – chyba mezi požadovaným výstupem a výstupem sítě

α_y^x [-] – výstup z neuronu

y_0 [-] – požadovaný výstup pro daný neuron

σ [-] – aktivační funkce

z_y^x [-] – suma vstupů do neuronu z předchozí vrstvy

$f(x)$ [-] – rovnice křivky

A, B, C [-] – konstanty určené z estimace pomocí `curve_fit` toolboxu

x [-] – proměnná reprezentující čas

9.2. Seznam zkratek

ANN (*Artificial Neural Network*) – umělá neuronová síť

AI (*Artificial Intelligence*) – umělá inteligence

NN (*Neural Network*) – neuronová síť

DFF (*Deep Feed Forward*) – vícevrstvé dopředné síť

BPFO (*Ball Pass Frequency Outer*) - frekvence při poškození vnějšího kroužku

BPFI (*Ball Pass Frequency Inner*) – frekvence při poškození vnitřního kroužku

BSF (*Ball Spin Frequency*) – frekvence při poškození valivých elementů

FTF (*Fundamental Train Frequency*) – frekvence při poškození klece

FFT (*Fast Fourier Transformation*) – rychlá Fourierova transformace

MLP (*Multi Layered Perceptron*) – vícevrstvý perceptron

RNN (*Recurent Neural Network*) – rekurentní neuronová síť

CNN (*Convolutional Neural Network*) – konvoluční neuronová síť

MSE (*Mean Square Error*) – Kvadratická hodnota chyby

MAE (*Mean Absolute Error*) – Absolutní hodnota chyby

RUL (*Remaining Usefull Life*) – Odhad zbývajícího života

UI – umělá inteligence

9.3. Seznam obrázků

[1.]	Stupně průmyslových revolucí	15
[2.]	Struktura neuronové sítě	17
[3.]	Struktura neuronu	17
[4.]	Struktura učení s učitelem	18
[5.]	Průběhy Aktivačních funkcí	19
[6.]	Průběhy křivek chybových funkcí	21
[7.]	Vliv velikosti kvantilu na proložení	21
[8.]	Schéma pro zpětnou propagaci	22
[9.]	Překonání plytkého lokálního minima gradientem se setrvačností	23
[10.]	Výstup sítě naučené příliš přesně na vzorek dat	25
[11.]	Nákres ložiska	27
[12.]	Výstup po prořezávání sítě	29
[13.]	Srovnání preventivní, reaktivní a prediktivní údržby	31
[14.]	Vzorek proložený na databázi poruch	32
[15.]	Threshold metoda a určení RUL	33
[16.]	Obálka FFT signálu pro vyhodnocení stavu ložiska	34
[17.]	Transformace 1D FFT na 2D pro konvoluční NN	34
[18.]	Struktura učení neuronové sítě	35
[19.]	Struktura odhadu RUL	36
[20.]	Uspořádání testovací sestavy	37
[21.]	Vliv ořezané periody na FFT	38
[22.]	FFT pro OK a NOK data	39
[23.]	Výsek z FFT obálky	40
[24.]	Vyhodnocení testovacího datasetu OK / NOK dat	41
[25.]	Nefiltrovaný a filtrovaný průběh degradace ložiska	42
[26.]	Jedna z možných struktur sítě	43
[27.]	Odhad degenerace ložiska	45
[28.]	Průběh života ložiska zobrazující rozdíl ve výskytu finální poruchy	46
[29.]	Importovaná vs doučená křivka pro Run-to-Failure metodu	47
[30.]	Možná chyba vzniklá při řešení polynomické rovnice pro RUL	48
[31.]	Řešení pro data s použitím jiné exponenciální křivky	49
[32.]	Odhad stavu a RUL součásti	50
[33.]	Výstup pro sít' groups	51
[34.]	Diskriminační matice	51
[35.]	Odhad stavu a RUL součásti (ložiska 1)	52
[36.]	Odhad stavu a RUL součásti (ložiska 2)	53
[37.]	Proložení odhadů stavů pro doučování modelu	54